REKURENS, METODE DAN APLIKASINYA UNTUK KOMPLEKSITAS WAKTU DAN *COMPLETE SEARCH*

Novan Parmonangan Simanjuntak/13509034
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
email : mathprince@ymail.com; 13509034@std.stei.itb.ac.id; sf_euler_novan@yahoo.com

ABSTRAK

Rekurens (recurrence) adalah masalah yang sering muncul dalam permasalahan kehidupan. Di bidang keilmuan apapun seperti Matematika,Informatika,Biologi, Fisika,Ekonomi dan lainnya. Secara sederhana , dalam Matematika rekurens adalah hubungan suku ke-n dengan suku sebelumnya. Di makalah ini akan dibahas permasalahan yang menggunakan rekurens(meliputi bagaimana ciri masalah rekurens serta mentranslasikannya ke dalam bentuk persamaan rekurens), teknik yang diperlukan untuk strategi pemecahan masalah, serta aplikasinya dalam bidang Informatika khususnya untuk Kompleksitas dan Complete Search. Hal dasar yang diperlukan adalah kemampuan koefisien binomial,enumerasi kombinatorika, kemampuan dasar matriks(seperti Eliminasi Gauss) dan sedikit teori bilangan mengenai modulo. Adapun enumerasi kombinatorika dan modulo akan dibahas sedikit di sini. Selain itu, berbagai aplikasi lain untuk graph,sumasi dan teori bilangan juga akan dibahas di sini untuk memperlihatkan bahwa enumerasi kombinatorika dan teknik rekurens merupakan salah satu basis pemecahan masalah dalam Matematika diskrit. Berbagai macam teknik akan dibahas di sini seperti penerapan enumerasi kombinatorika,deret telescoping, fungsi pembangkit, serta metode permisalan. Aplikasi yang terutama akan dibahas di sini mengenai pemakaian rekurens untuk kompleksitas waktu dan complete search.

Kata kunci : Rekurens, Enumerasi Kombinatorika, Kompleksitas Waktu, Complete Search.

1. PENDAHULUAN

Begitu banyak permasalahan yang muncul ketika kita berhadapan dengan benda diskrit, Banyak teknik yang dipakai untuk menyelesaikan masalah diskrit, seperti tori bilangan, induksi matematika, enumerasi kombinatorika, prinsip sangkar merpati, graph, persegi latin dan salah satunya adalah rekurens. Rekurens sering muncul dalam permasalahan akan tetapi kebanyakan hanya tersirat saja. Pembahasan rekurens di sini hanyalah untuk pengenalan saja, diharapkan pembaca dapat mendapatkan gambaran mengenai rekurens, sebab rekurens dipandang sebagai jembatan antara ilmu Matematika dengan ilmu lainnya selain sebagai jembatan Matematika Diskrit dan Matematika Kontinu.

2. REKURENS(*RECURRENCE*)

2.1 Definisi dan Penggunaan

Dalam matematika rekurens berarti hubungan dalam barisan dengan bentuk

$$a_n = f(a_{n-1},a_{n-2},\ldots,a_{n-p}) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(1)$$

Perhatikan bahwa nilai p tergantung pada nilai n dan nilai awal(basis) diketahui. Nilai rekurens dapat berupa linear dan non-linear, ini tergantung pada fungsi f. Untuk rekurens linear, bentuk di atas dapat ditulis sebagai

$$a_n = c_1a_{n-1} + c_2a_{-2} + \ldots + c_pa_{n-p} ,\ldots\ldots\ldots\ldots\ldots..(2)$$

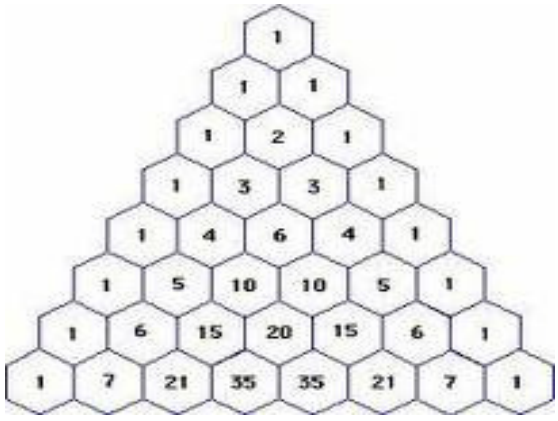
dengan p dan $c_1c_2\ldots c_p$ adalah konstanta, dan nilai awal(basis) $a_0,a_1,\ldots,a_{p-1}$ diketahui.

Dari definisi di atas terlihat bahwa rekurens merupakan hubungan unik yang mendefinisikan barisan di mana kita bisa menghitung subbarisan berikutnya dengan basis dan fungsi rekursif yang diberikan.

Berikut akan diberikan deskripsi sederhana untuk menggambarkan rekurens dan memperoleh hubungan rekurens :

Deskripsi 1 : Deret Bilangan Fibonacci 0,1,1,2,3,5,8,13,21… . Awalnya kita tidak langsung diberitahu bahwa terdapat rekurens pada permasalahan barisan ini. Adapun rekurens di sini yaitu :
Nilai awal (basis) : $F_0=0$; $F_1=1$;
Fungsi rekurens : $F_n=F_{n-1}+F_{n-2}$;

Deskripsi 2 : Pada segitiga pascal di bawah ini :



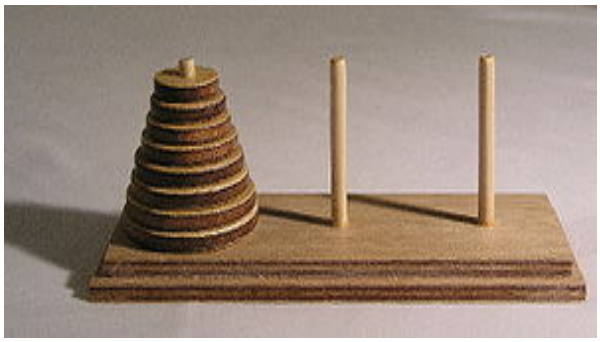
Misalkan kita menyatakan baris pertama sebagai $p_{0,0}$,

Kemudian baris kedua sebagai $p_{1,0}$ dan $p_{0,1}$ secara umum kita bisa nyatakan secara umum untuk baris berikutnya sebagai

$$p_{n,0},\ p_{(n-1),1},\ p_{(n-2),2},\ \cdots\ p_{1,(n-1)},\ p_{0,n}$$

Untuk kasus segitiga pascal kita dapat relasi rekurensnya yaitu bahwa nilai suatu bilangan merupakan penjumlahan dari kedua bilangan di atasnya.
Rekurensnya di sini yaitu :
Nilai awal(basis) : $\mathbf{p_{n,0}=0;\ p_{0,n}=0}$;
Fungsi rekurens : $\mathbf{p_{j,k} = p_{(j-1),k} + p_{j,(k-1)}}$

Deskripsi 3 : Permasalahan menara Hanoi
Permasalahan menara Hanoi merupakan salah satu masalah yang menggunakan rekurens, di sini diketahui bahwa terdapat tumpukan(stack) cincin dengan jari-jari menurun ke atas dan 3 batang. Pertanyaanya adalah berapa total minimum cara memindahkan keseluruh cincin tersebut ke batang yang ketiga(batang kedua dapat digunakan sebagai perantara) dengan syarat cincin yang jari-jarinya lebih kecil harus diletakkan di atas cincin yang jari-jarinya lebih besar. Berikut gambaran menara Hanoi:



Misalkan kita menyatakan t_n sebagai langkah minimum yang diperlukan untuk n cincin. Di sini jelas bahwa nilai minimum untuk 1 cincin adalah 1.Sehingga didapat nilai m_1=1.Sekarang misalkan kita ingin mencari total langkah minimum untuk n cincin,yaitu m_n, dari sini kita bisa menggunakan nilai m_{n-1} yang merupakan langkah minimum untuk n-1 cincin. Untuk memindahkan n cincin, kita perlu memindahkan n-1 cincin ke tengah (m_{n-1} langkah) kemudian pindahkan cincin terbawah ke batang ketiga(1 langkah) dan yang terakhir memindahkan (n-1) cincin ke batang ketiga($m_{n,1}$) langkah.Dibutuhkan $2m_{n-1}+1$ langkah untuk n cincin. Kita dapat rekurensnya :
Nilai awal(basis) : $m_1=1$;
Fungsi rekurens : $m_n=2m_{n-1}+1$

Deskripsi di atas merupakan contoh masalah yang menggunakan rekurens . Itu gampang untuk kita mendapatkan fungsi rekurensnya. Akan tetapi sangatlah sulit untuk menetukan nilai fungsi dari dari n secara langsung.Berikut ini beberapa teknik yang akan dibahas dalam rekurens :
1.Deret Telescoping
2.Fungsi Pembangkit(Generating Function)
3.Metode Permisalan
Adapun untuk mampu menggunakan teknik rekurens ada beberapa syarat yang perlu dikuasai, yaitu :
1.Enumerasi Kombinatorika
2.Modulo
Berikut ini merupakan contoh aplikasi dari rekurens :
1.Menghitung nilai kompleksitas waktu (T(n)) dari sebuah algoritma.
2.Mengaplikasikan rekurens untuk teknik penyelesaian Compelete Search
3.Di bidang Biologi, yaitu untuk pemodelan proses pertumbuhan populasi.
4.Di bidang Elektro, yaitu untuk prosesing signal digital.
5.Di bidang Ekonomi, yaitu untuk pemodelan sektor ekonomi dan analisis deret waktu.
Adapun yang akan kita bahas di sini yaitu untuk no.1(Kompleksitas waktu) dan no.2(Complete Search),
Untuk no 3,4 dan 5 tidak akan dibahas di sini karena membutuhkan definisi awal dan konsep awal mengenai bidang keilmuan masing-masing.Akan tetapi konsep rekurens ini akan sangat berguna untuk pemrograman untuk fasilitas bidang keilmuan lain.
2.2 Konsep awal yang diperlukan

Dalam mengaplikasikan rekurens untuk kompleksitas waktu dan complete search diperlukan kemampuan fungsi modulo dan juga enumerasi kombinatorika , adapun di sini akan dibahas sedikit saja.
2.2.1 Aritmatika modulo dan Kekongruenan

2.2.1.1 Aritmatika modulo

Untuk a, m, bilangan bulat, notasi modulo yaitu

a mod m = r sedemikian sehingga a=mq+r,untuk $0 \le r < m$.

Contoh :
14 mod 23 = 14. (14=23.0+14)
-6 mod 4 = 2 (-6=4/2+2)

2.2.1.2 Kekongruenan

Untuk a,b,m bilangan bulat,
$a \equiv b \pmod m$ berarti m membagi (a-b)
dapat ditulis menjadi a=b+km untuk k bilangan bulat.

Contoh :
$12 \equiv 5 \pmod 7$.12=5+7.1
2.2.2 Enumerasi Kombinatorika

Di sini akan dibahas cara memanfaatkan enumerasi kombinatorika secara optimal, ada banyak permasalahan yang bisa dipecahkan oleh kombinatorika(solusi alternative).Tetapi karena hanya “tersirat” saja, maka sangat sulit menemukan solusi.
Kita di sini hanya akan menggunakan definisi perkalian, penjumlahan dan kombinasi r elemen dari n elemen, berikut deifinisi untuk kombinasi

$$\binom{n}{k} = \frac{n^{\underline{k}}}{k!} = \frac{n(n-1)(n-2)\cdots(n-k+1)}{k(k-1)(k-2)\cdots 1} = \prod_{i=1}^{k} \frac{n-k+i}{i}.$$

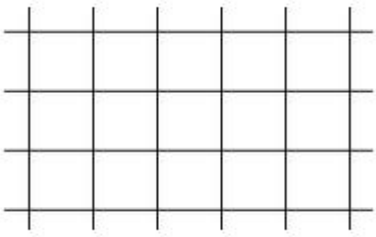
Definisi di atas menunjukkan banyaknya cara untuk memilih r elemen dari n elemen.
Mulai dari sini n combinasi r ditulis (n,r).

Contoh 1 :
Buktikan bahwa
$(n,0) + (n,1) + (n,2) + \ldots + (n,n) = 2^n$.

Solusi :
Solusi di atas mudah diselesaikan dengan koefisien binomial, akan tetapi terdapat solusi alternatif dengan menggunakan enumerasi kombinatorika. Di sini kita mengartikan (n,r) sebagai banyak subset elemen,sehingga (n,0) adalah subset dengan 0 elemen, (n,1) adalah subset dengan 1 elemen dan seterusnya. Karena dijumlahkan, maka semua subset tersebut memberntuk total subset(yang banyaknya 2^n).

Contoh 2 :
Perhatikan gambar grid di bawah ini :

B



A

Berapa banyak rute berbeda untuk langkah minimum dari A(titik paling kiri bawah) ke B(titik paling kanan atas)?

Solusi :
Untuk menghitung banyaknya cara terpendek dari A ke B sama saja dengan banyak cara memilih 3 grid dari 8 grid atau 5 grid dari 8 grid,
Misalkan garis satuan menyatakan satu langkah, maka dari A ke B terdapat 8 (3+5) langkah, dari sini kita akan menghitung banyaknya kemungkinan langkah tegak(sebanyak 3 buah) disusun di dalam ke delapan langkah, sehingga total langkah adalah (8,3) jalan.

Hal ini juga berlaku untuk banyak cara menyusun langkah mendatar(ada 5 langkah) ke dalam delapan langkah ini,didapat total langkah (8,5) jalan.
Jawaban solusi : (8,3)=(8,5)=56 rute.

Contoh 3 :
Buktikan bahwa
$(n,0)^2+(n,1)^2+(n,2)^2+\ldots+(n,n)^2=(2n,n)$

Solusi :
Di sini kita tidak akan menggunakan aljabar, tetapi menggunakan enumerasi kombinaatorika.
(2n,n) menyatakan banyaknya cara memilih n orang dari himpunan 2n orang yang terdiri dari n orang pria dan n orang wanita, maka cara memilih ini juga dapat dinyatakan dengan banyaknya cara memilih k pria dan (n-k) pria untuk k dari 0 sampai n.
Kita akan menggunakan sifat :

$(n,k) = (n, n-k)$, untuk $0 \leq k \leq n$.

Untuk kasus pemilihan tanpa priadan n wanita :
$(n,0) * (n,n) = (n,0)*(n,0) = (n,0)^2$
Untuk kasus pemilihan 1 pria dan (n-1) wanita :
$(n,1) * (n,n-1) = (n,1)*(n,1) = (n,1)^2$
…
…
…
Untuk kasus pemilihan n pria dan tanpa wanita :
$(n,n) * (n,0) = (n,n)*(n,n) – (n,n)^2$

Total pemilihan :
$(n,0)^2+(n,1)^2+(n,2)^2+\ldots+ n,n)^2$
Terbukti.

Contoh 4 (Pemibuktian teori bilangan dengan enumerasi kombinatorika) :

Buktikan Teorema Kecil Fermat :
Misalkan a adalah bilangan bulat positif dan p merupakan bilangan prima. Maka

$a^p \equiv a \bmod p$

Solusi :
Misalkan kita mempunyai mutiara dengan a buah warna berbeda.Dari sini kita buat kalung dengan p mutiara.Pertama kita membuat string mutiara(deretan mutiara yang belum disambung kedua ujungnya untuk membentuk kalung). Dapat dihitung bahwa jumlah kemungkinan penyusunan string tersebut adalah sebanyak a^p. Jika kita membuang kemungkinan string dengan 1 warna saja,maka sisa kemungkinan string ada sebanyak a^p-a. Sekarang kita sambungkan kedua ujung setiap kemungkinan string untuk membentuk kalung. Perhatikan bahwa 2 string berbeda hanya oleh permutasi siklis dari mutiara tersebut sehingga menghasilkan kalung yang tidak dapat dibedakan. Tetapi ada sebanyak p permutasi siklis.Karena jumlah kalung yang berbeda adalah $(a^p-a)//p$ merepresentasikan bilangan bulat, terbukti bahwa

$p \mid a^p-a$

dengan kongruensi,didapat :

$a^p-a \equiv 0 \bmod p$
$a^p \equiv a \bmod p$
$a^{p-1} \equiv 1 \bmod p$

2.3 Teknik penyelesaian permasalahan rekurens

Ada banyak teknik yang dapat digunakan untuk menyelesaikan maslah rekurens, di makalah ini akan dibahas metode deret telescoping , fungsi pembangkit dan metode permisalan.

2.3.1 Deret Telescoping

Deret telescoping adalah deret yang jumlahnya bisa ditemukan dengan menggunakan hubunngan antar suku yang bisa saling menyederhanakan.

Contoh 5 :

Hitung hasil penjumlahan deret

$$\sum_{n=1}^{\infty} \frac{1}{n(n+1)}$$

Perhatikan bahwa didapat fungsi rekurens
f(n+1)= (n/n+2).f(n)
Persamaan ini dapat diselesaikan dengan saling menyederhanakan suku n dengan suku-n+1,
Penjumlahan deret di atas dapat ditulis

$$\sum_{n=1}^{\infty} \frac{1}{n(n+1)} = \sum_{n=1}^{\infty} \left(\frac{1}{n} - \frac{1}{n+1} \right)$$

$$= \left(1 - \frac{1}{2}\right) + \left(\frac{1}{2} - \frac{1}{3}\right) + \cdots$$

$$= 1 + \left(-\frac{1}{2} + \frac{1}{2}\right) + \left(-\frac{1}{3} + \frac{1}{3}\right) + \cdots = 1$$

Suku terakhir deret di atas tidak harus untuk penjumlahan suku samapai tak-hingg(bisa untuk suku ke-n, dengan n ≥1, n bilangan bulat).
Berikut untuk penjumlahan sampai suku ke-k,dengan sembarang bilangan bulat > 1.

$$\sum_{n=1}^{k} \frac{1}{n(n+1)} = \sum_{n=1}^{k} \left(\frac{1}{n} - \frac{1}{n+1} \right)$$

$$= \left(1 - \frac{1}{2}\right) + \left(\frac{1}{2} - \frac{1}{3}\right) + \ldots + \left(\frac{1}{k} - \frac{1}{k+1}\right)$$

$$= 1 + \left(-\frac{1}{2} + \frac{1}{2}\right) + \ldots + \left(-\frac{1}{k} + \frac{1}{k}\right) + \left(-\frac{1}{k+1}\right)$$

$$= \frac{k}{k+1}$$

Contoh 5 :
Tentukan ni;ai

$$\sum_{n=1}^{k} n(n!)$$

Solusi :

$$\sum_{n=1}^{k} n(n!) = \sum_{n=1}^{k} ((n+1)-1)(n!)$$

$$= \sum_{n=1}^{k} ((n+1)! - n!)$$

$$= (2! - 1!) + (3! - 2!) + \ldots + (n! - (n-1)!) + ((n+1)! - n!)$$

$$= (-1!) + (2! - 2!) + (3! - 3!) + \ldots + (n! - n!) + (n+1)!$$

$$= (n+1)! - 1$$

Ide telescoping akan digunakan juga dalam fungsi pembangkit (untuk menyederhanakan fungsi).

2.3.2 Fungsi Pembangkit

Di makalah ini akan dibahas cara menggunakan fungsi pembangkit secara umum saja, untuk mteode syarat seperti metode pecahan parsial dan deret tak hingga tidak dibahas di sini.

Misalkan $a_0,a_1,a_2\ldots$ merupakan deret, kita menulis

$$G(a_n; x) = \sum_{n=0}^{\infty} a_n x^n.$$

\

Deret ini disebut fungsi pembangkit biasa untuk barisan yang diberikan.
Berikut contoh pemakaian yang umum,yaitu untuk menetukan nilai F_n(suku ke n dari deret bilangan Fibonacci).

Contoh 6 :Tentukan rumus umum ke-n dari deret bilangan Fibonacci, di mana

$a_0=0,a_1=1;$ $a_n=a_{n-1}+a_{n-2}$

Solusi :
Buat fungsi pembangkit barisan bilangan Fibonacci, yaitu :

$G(x) = a_0+a_1x+a_2x^2+ a_3x^3+ a_4x^4+ a_5x^5+...$

Di sini kita akan cari nilai G(x) dengan menggunakan teknik telescoping, perhatikan bahwa
$a_n=a_{n-1}+a_{n-2}$,
Kita akan memanfaatkan sifat di atas untuk saling menghilangkan suku-suku di G(x)

$G(x) = a_0+a_1x+a_2x^2+ a_3x^3+ a_4x^4+ a_5x^5+...$

$xG(x) = a_0x+a_1x^2+ a_2x^3+ a_3x^4+ a_4x^5+...$

$x^2G(x) = a_0x^2 + a_1x^3 + a_2x^4 + a_3x^5 + ...$

$G(x) - xG(x) - x^2G(x) = a_0 + (a_1 - a_0)x + (a_2 - a_1 - a_0)x^2 + (a_3 - a_2 - a_1)x^3 + \ldots$

$(1 - x - x^2).G(x) = 0 + (1-0)x + 0.x^2 + 0.x^3 + \ldots$

$(1 - x - x^2).G(x) = x$

$G(x) = \frac{x}{1 - x - x^2}$

Perhatikan bahwa G(x) dapat ditulis menjadi

$$G(x) = \frac{1}{\sqrt{5}}\left(\frac{1}{1 - \left(\frac{1+\sqrt{5}}{2}\right)x} - \frac{1}{1 - \left(\frac{1-\sqrt{5}}{2}\right)x}\right)$$

Kita bisa memisalkan

$$p = \left(\frac{1+\sqrt{5}}{2}\right) \quad \text{dan} \quad q = \left(\frac{1-\sqrt{5}}{2}\right)$$

Sehingga bisa ditulis

$$G(x) = \frac{1}{\sqrt{5}}\left(\frac{1}{1 - px} - \frac{1}{1 - qx}\right)$$

Dari definisi deret tak hingga ,

$$\frac{1}{1 - px} = 1 + px + p^2x^2 + p^3x^3 + \ldots$$

$$\frac{1}{1 - qx} = 1 + qx + q^2x^2 + q^3x^3 + \ldots$$

Didapat

$$G(x) = \frac{1}{\sqrt{5}}\left((1 + px + p^2x^2 + p^3x^3 + \ldots) - (1 + qx + q^2x^2 + q^3x^3 + \ldots)\right)$$

$$G(x) = \frac{1}{\sqrt{5}}\left((1-1) + (p-q)x + (p^2 - q^2)x^2 + \ldots\right)$$

Perhatikan bahwa dari deret terakhir di atas didapat

$a_n = (p^n - q^n)//5$

yang merupakan suku ke-n dari deret bilangan Fibonacci.

2.3.3 Metode permisalan.

Metode ini merupakan salah satu metode yang mudah digunakan, metode yang diperlukan hanyalah metode eleminasi dan pengetahuan tentang berbagai macam bentuk fungsi.Selain untuk menggunakan untuk menetukan fungsi rekurens,kita juga bisa gunakan untuk menetukan deret.

. Misalkan diberikan suatu deret

Berikut langkah-langkah dalam metode permisalan :

1. Taksir dan buat bentuk umum yang mungkin untuk Rumus umum f_n. Fungsi f_n ini bisa ditebak dengan Melihat kecenderungan perubahan suku , bisa linear(berbentuk ax+b), polinomial($a_0+a_1x+a_2x^2+\ldots$), eksponensial(ax^n), dan lainnya.
2. Cari formula untuk f_n dengan mencari nilai tiap Koefisien dan konstanta dari f_n dengan eliminasi Dan subsitutusi variable dari deret yang diberikan.

Metode ini bisa dibilang tidak “menjamin”.Ini karena sangatlah sulit untuk mendapatkan fungsi suatu bilangan di luar polinomial,terutama eksponensial.Di makalah ini akan dibahas untuk bentuk polynomial saja. Berikut contoh dari penerapan metode permisalan.

Jika diberikan deret polinomial berderajat d(suku ke-n berbentuk polynomial), maka jumlah dari deret sampai suku ke-k adalah polynomial dengan variabel k dan derajat /derajat d+1.

Contoh 7:

Tentukan nilai dari

$$\sum_{n=1}^{k} n^2$$

Solusi :

Misalkan

$$f(k) = \sum_{n=1}^{k} n^2$$

Maka f(k) berderajat 3(karena derajat $U_n = 2$).

Maka kita bisa memisalkan

$f(k) = ak^3 + bk^2 + ck + d$

Perhatikan bahwa terdapat 4 konstanta yang harus dicari (a,b,c,d), maka kita perlu 4 persamaan.

Kita akan mencari nilai a,b,c,d dengan memanfaatkan f(1),f(2),f(3) dan f(4).

$f(1) = a + b + c + d = 1$ ……1

$f(2) = 8a+4b+2c+d = 5$ ...……...2

$f(3) = 27a+9b+3c+d=14$……..3

$f(4)=64a+16b+4c+d=30$……..4

Eliminasi keempat persamaan tesebut, didapat

$$a = \frac{1}{3}, b = \frac{1}{2}, c = \frac{1}{6}, d = 0$$

Sehingga nilai penjumlahan dari deret sampai suku-k :

$$f(k) = \frac{1}{3}k^3 + \frac{1}{2}k^2 + \frac{1}{6}k$$

.

3 Aplikasi rekurens untuk kompleksitas waktu dan complete search.

3.1 Aplikasi rekurens untuk kompleksitas waktu.

Menetukan kompleksitas waktu dari algoritma yang dilakukan adalalah sesuatu yang wajib.Biasanya orang diajarkan untuk menghitung O(n). Banyak metode yang bisa dipakai , salah satunya menggunakan ketidaksamaan, atau dengan mencarinilai T(n) terlebih dahu;u. Tentu saja cara ketidaksamaan lebih efisien, akan tetapi jika kita bisa menemukan nilai dari T(n), tentu kita bisa dengan yakin menetukan nilai O(n), karena T(n) lebih mencerminkan sekompleks apa suatu algoritma .Misalkan saja kita ingin menghitung kompleksitas tercepat dari 2 algoritma.Jika kedua algoritma mempunyai nilai O(n) yang sama , tentu kita tidak tahu mana algoritma yang lebih mangkus.Metode untuk memastikannya adlaah dengan menetukan nilai T(n) itu sendiri.Di sini kita aka menghitung nilai kompleksitas waktu (T(n)).Rekurens dapat dimanfaatkan untuk kompleksitas waktu, tentu ini karena kemampuan dalam deskkripsi dan pemecahan masalah rekurens dapat membantu mencari solusi dari kompleksitas waktu, berikut diberikan contoh mudah untuk menghitung kompleksitas waktu.

Contoh 8 :
Hitung kompleksitas waktu (T(n)) dari algoritma berikut

```
include <iostream>
using namespace std;

int main()
{
   int i,j,t=0,n;
   cin >> n;
   for(i=1;i<=n;i++)
     for(j=1;j<=i*i;j++)
       t++;
   cout << t;
   system("PAUSE");
   return 0;
}
```

Solusi :

Dari algoritma di atas didapat
$T(n) = 1^2+2^2+\ldots+n^2$

Karena U_n adlaah polynomial,maka akan diselesaikan dengan metode permisalan. Didapat T(n) adalah polynomial berderajat 3.
Kita misalkan $T(n)=an_3+bn^2+c^n+d$
Lalu dengan eliminasi :

T(1) = a + b + c + d = 1 …….1

T(2) = 8a+4b+2c+d = 5 ...……...2

T(3) = 27a+9b+3c+d=14……..3

T(4)=64a+16b+4c+d=30……..4

Eliminasi keempat persamaan tesebut, didapat

$$a = \frac{1}{3}, b = \frac{1}{2}, c = \frac{1}{6}, d = 0$$

Kita dapat nilai T(n) :

$$T(n) = \frac{1}{3}n^3 + \frac{1}{2}n^2 + \frac{1}{6}n$$

3.2 Aplikasi rekurens untuk complete search

Rekurens digunakan di complete search untuk menyederhanakan suatu solusi algoritma dari banyaknya kemungkinan yang ada. Di sini kita menggunakan prinsip perulangan suatu kemungkinan yang terjadi , di sini akan dibahas persoalan mengenai complete search.
Berikut deskripsi persoalan mengenai complete search.

Deskripsi 4 :
Misalkan diberikan N($1 \le N \le 10000$) buah lampu secara berurutan keadaan aal N buah lampu(menyala atau tidak) dan 4 tombol.Tombol pertama menyalakan/mematikan semua lampu. Lampu yang kedua untuk lampu urutan genap Yang ketiga untuk lampu ganjil.Dan terakhir untuk lapu dengan urutan 1,4,7,10,…/Keluarkan semua kemungkinan state yang mungkin dialami oleh N buah lampu tersebut.

Perhatikan kita bisa menghitung kompleksitas waktu secara kasar untuk suatu algoritma dengan menggunakan enumerasi kombinatorika. Misalkan kita ingin menggunakan teknik coba-coba/*brute-force* mencoba keempat kemungkinan, maka total kemungkinan yang ingin kita coba adalah sebanyak 4^{10000}, artinya kita tidak mungkin melakukan cara coba-coba.
Sekarang perhatikan bahwa urutan dari tombol yang ditekan tidak diperhatikan , ini berarti kita harus mencoba 10000^4 total kemungkinan.Banyak kemungkinan ini masih sangat besar.
Sekarang perhatikan bahwadengan prinsip rekurens, menekan tombol sebanyak dua kali sama saja dengan tidak menekan tombol sama sekali.Sehingga kita hanya perlu mengecek untuk tombol tidak ditekan dan untuk tombol ditekan sebanyak 1 kali. Total kemungkinan

hanyalah 2^4=16 kemungkinan. Tentu saja ini dengan mudah dapat diselesaikan.

Deskripsi 5 :
Ada 9 buah jam dengan hanya bisa menunjuk ke angka 12:00,3:00,6:00 atau 9:00/ Tujuan kita adlah untuk memanipulasi kesembilan jam ini agar semuanya menunjuk jam 12:00.Satu-satunya cara untu memanipulasi jam ini adalah dengan menggunakan satu dari kesembilan tipe rotasi. Masing-masing merotasikan subhimpunan tertentu dari jam dengan 90 derajat searah jarum jam. Temukan barisan untuk tercepat gerakan yang membuat semua jam menunjuk ke angka 12:00.

Hal yang jelas yang harus dilakukan adalah dengan menggunakan solusi rekursif, yang mengecek untuk melihat apakah ada solusi dengan 1 gerakan, 2 gerakan dan sebagainya. Secara kasar , dengan menggunakan enumerasi kombinatorika ada 9^9 kemingkinan, di mana k adalah banyaknya gerakan. Karena nilai k(jumlah gerakan) bisa banyak besar sekali, maka cara ini tidak mangkus untuk diimplementasikan.
Perhatikan bahwa urutan dari gerakan tidaklah masalah. Sehingga ada sekitar k^9 kemungkinan yang harus dicoba, akan tetapi ini masih belum cukup.
Sekarang perhatikan bahwa melakukan 4 gerakan sama saja dengan tidak menggunakan gerakan sam sekali(kembali ke posisi awal jam),ini berarti kita tidah menggunakan gerakan lebih dari 3 kali. Oleh karena itu terdapat 4^9 =262072 kemungkinan yang harus dicoba. Tentu ini bisa diselesaikan komputer dengan cepat.

Banyak sekali manfaat dari rekurens untuk bidang keilmuan, baik informatika dan di luar informatika.Diharapkan dengan makalah ini, pembaca dapat terbukan pikirannya untuk memahami rekurens, metode dan aplikasinya.

4. KESIMPULAN

Dari semua penjelasan mengenai rekurens,metode dan aplikasinya , bisa kita tarik kesimpulan :

1. Prinsip rekurens sangat penting, karena bisa diterapkan untuk benda diskrit dan benda tidak diskrit.
2. Prinsip rekurens bermanfaat dalam perhitungan jumlah, relasi antar suku di dalam deret.
3. Pemahaman prinsip rekurens membutuhkan tools dari bahasan lain seperti kombinatorika dan teori bilangan.
4. Rekurens punya banyak aplikasi, permaslahan yang sulit adalaa bagaimana merepresentasikan masalah ke dalam bentuk rekurens sehingga bisa dipecahkan.

REFERENSI

[1] M.Aigner: *Combinatorial theor(*(Springer-Verlag,1979)
[2] L.Mirsky:*Traversal theory* (Academic Press)
[3] http://en.wikipedia.org/wiki/Recurrence_relation
diakses tanggal 10 Desember 2010, jam 22:00
[4] Engel,Arthur : *Problem-Solving*-Strategies (Springer Verlag)

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 29 April 2010

ttd

Novan Parmonangan Simanjuntak
13509034