

Penggunaan Algoritma Backtracking dalam Pencarian Solusi Fubuki Puzzle

Wulandari (13506001)

Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
Jln. Ganeca no. 10 Bandung Indonesia
e-mail: if16001@students.if.itb.ac.id

ABSTRAK

Dalam kehidupan sehari-hari, kita sebagai manusia sering menemukan permasalahan-permasalahan yang membutuhkan penyelesaian. Dalam beberapa persoalan, penyelesaian tidak hanya dihasilkan oleh satu cara, namun terdapat beberapa cara untuk mendapatkan solusi yang diinginkan. Tentunya tiap cara tersebut memiliki tingkat keefektifan dan efisiensi yang berbeda-beda, sehingga kita harus dapat menentukan solusi mana yang dapat menyelesaikan permasalahan seefektif dan seefisien mungkin. Salah satu contohnya adalah dalam hal penyelesaian permainan Fubuki – *Sebuah permainan yang berasal dari Jepang dimana pemain diminta untuk mengisi kotak dengan angka yang sesuai.* Dalam menyelesaikan permainan ini, algoritma yang dirasa paling efektif dan efisien adalah algoritma Backtracking. Algoritma yang berbasis pada DFS ini merupakan perbaikan dari algoritma brute-force yang membuat kita tidak perlu memeriksa semua kemungkinan yang ada, melainkan hanya memeriksa kemungkinan yang mengarah ke solusi saja. Akibatnya, waktu pencarian dapat diterima.

Kata kunci: Backtracking, Fubuki.

1. PENDAHULUAN

1.1 Algoritma Backtracking

Istilah Backtracking diperkenalkan pertama kali oleh D.H. Lehmer pada tahun 1950. Selanjutnya, R.J. Walker, Golomb, dan Baumert menyajikan uraian umum tentang algoritma ini dan penerapannya pada berbagai persoalan. [HOR78].

Untuk menerapkan metode runut balik, properti berikut didefinisikan :

a. Solusi persoalan

Solusi dinyatakan sebagai vektor dengan n -tuple :

$$X = (x_1, x_2, x_3, \dots, x_n)$$

b. Fungsi pembangkit nilai

Dinyatakan sebagai : $T(k)$

c. Fungsi pembatas

Dinyatakan sebagai : $B(x_1, x_2, x_3, \dots, x_n)$

1.2 Fubuki Puzzle

Fubuki adalah sebuah permainan yang berasal dari Jepang dimana pemain diminta untuk mengisi kotak dengan angka yang sesuai. Di bawah ini adalah salah satu contoh soal Fubuki Puzzle.



Gambar 1. Contoh Soal Fubuki

Untuk menyelesaikan persoalan ini, pemain harus mengisi sembilan kotak yang belum terisi dengan aturan sebagai berikut :

- Masing – masing kotak berisi sebuah angka dari 1 hingga 9
- Tidak ada kotak yang berisi angka yang sama. Ini berarti semua angka dari satu hingga sembilan diisikan tepat di satu kotak
- Jika angka – angka yang berada pada baris yang sama dijumlahkan, maka akan menghasilkan jumlah sesuai dengan kotak yang ada di paling kanan setiap barisnya.
- Jika angka – angka yang berada pada kolom yang sama dijumlahkan, maka akan menghasilkan jumlah sesuai dengan kotak yang ada di paling bawah setiap barisnya.

e. Tiap soal hanya memiliki satu buah solusi permasalahan.

Sesuai dengan aturan di atas, maka soal yang terdapat pada Gambar 1 bila dicari solusinya akan mengeluarkan hasil seperti gambar di bawah ini.

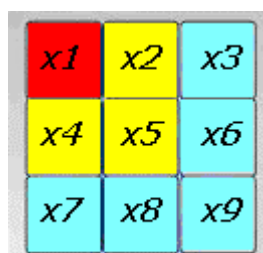


Gambar 2. Solusi Soal Fubuki

3. PENYELESAIAN MENGGUNAKAN BACKTRACKING

Jika setiap kotak yang harus diisi diberi variabel x , maka kita akan mendapatkan solusi persoalan berupa $X = (x_1, x_2, x_3, \dots, x_n)$ dimana $x_1, x_2,$ dan x_3 merupakan kotak yang berada pada baris pertama, $x_4, x_5,$ dan x_6 merupakan kotak yang berada pada baris kedua, dan $x_7, x_8,$ serta x_9 merupakan kotak yang berada pada baris ketiga terurut dari kiri ke kanan. Ruang solusinya adalah semua permutasi dari angka – angka 1, 2, 3, 4, 5, 6, 7, 8, 9.

Penyelesaian dimulai dengan mengisi 1 pada x_1 . Setelah itu diteruskan dengan mengisi kotak di sebelahnya dengan mempertimbangkan apakah angka tersebut sudah digunakan sebelumnya dan apakah dengan ditamhkannya angka tersebut masih memungkinkan untuk mencapai hasil penjumlahan sesuai dengan yang telah ditentukan oleh soal. Oleh karena itu, kita membagi sembilan kotak tersebut menjadi 3 bagian.



Gambar 3. Pembagian Pemrosesan

- Bagian pertama, yang berwarna merah, merupakan bagian yang tidak memerlukan pengecekan apapun.
- Bagian kedua, yang berwarna kuning, merupakan bagian yang memerlukan pengecekan apakah nomer yang ingin diisikan sudah ada pada kotak sebelumnya dan jika nomer tersebut diisikan pada kotak, apakah masih ada kesempatan untuk memperoleh jumlah yang sesuai. Jika kedua persyaratan tersebut tidak terpenuhi, maka akan dilakukan runut balik.
- Bagian ketiga, yang berwarna biru, merupakan bagian yang memerlukan pengecekan seperti yang dilakukan bagian kedua, sekaligus mengecek apakah jumlah angka yang berada pada satu baris atau pada satu kolom yang sama memiliki jumlah yang benar. Hal ini dikarenakan oleh kotak yang berwarna biru berada di kolom terkanan atau baris terbawah di antara kesembilan kotak tersebut. Ini berarti pengecekan penjumlahan dilakukan pada bagian ini. Jika ketiga persyaratan tersebut tidak terpenuhi, maka akan dilakukan runut balik.

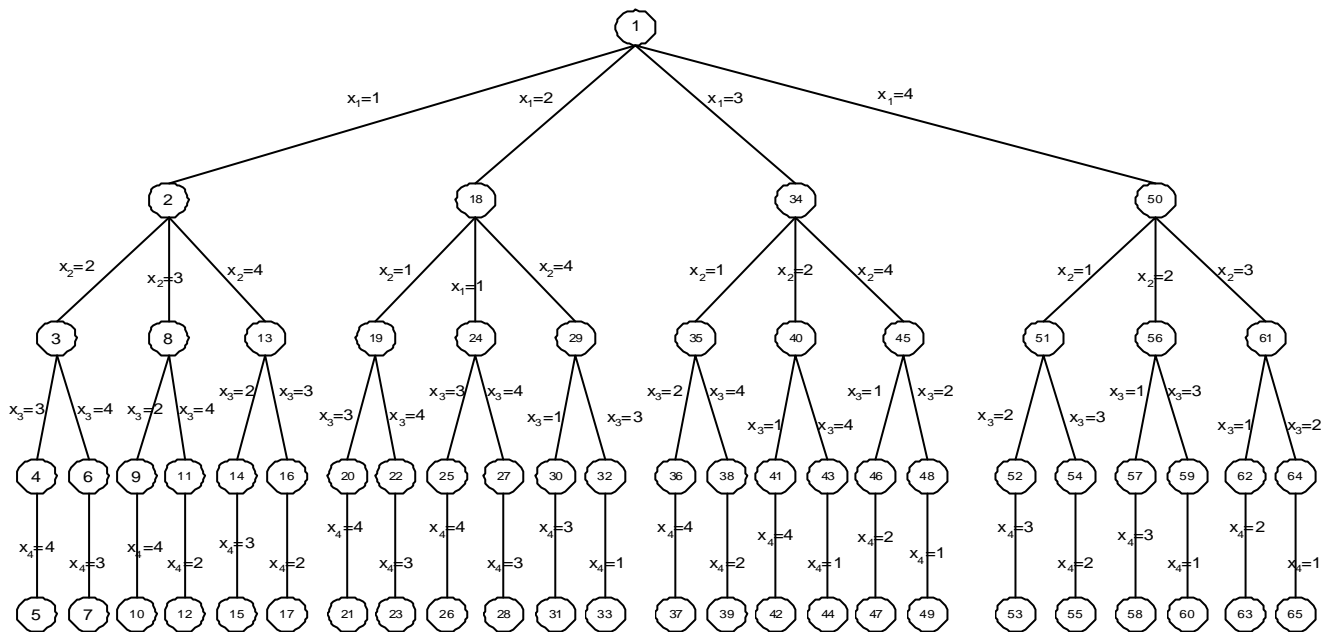
Untuk merealisasikan hal yang telah dijelaskan di atas, pada algoritma ini diperlukan beberapa fungsi tambahan, diantaranya :

- Cek_awal : fungsi ini digunakan untuk mengecek apakah jika angka diisikan ke dalam kotak, masih ada kemungkinan untuk memenuhi penjumlahan. Hal ini berarti, jumlah sementara yang dicapai harus lebih kecil dari jumlah yang seharusnya. Hal ini berlaku pada setiap baris maupun kolom. Fungsi ini menerima masukan angka yang ingin diisi dan x yang merupakan array of integer.
- Cek_akhir : fungsi ini digunakan untuk mengecek apakah jika angka diisikan ke dalam kotak, akan menghasilkan jumlah yang sesuai dengan yang diinginkan oleh soal. Hal ini berarti fungsi ini akan menjumlahkan angka – angka yang berada pada baris yang sama atau pada kolom yang sama bergantung pada oleh siapa fungsi ini dipanggil. Fungsi ini menerima masukan angka yang ingin diisi dan x yang merupakan array of integer.

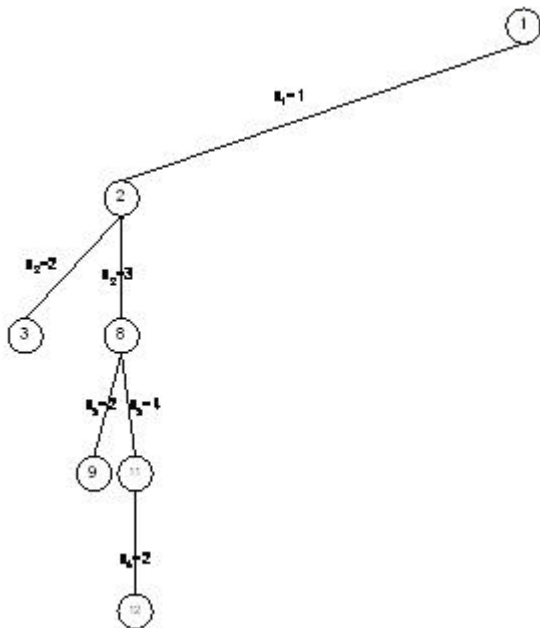
Kita akan mencoba untuk menyelesaikan satu permasalahan Fubuki Puzzle ini. Namun, untuk menyederhanakan permasalahan, kita akan mencoba menyelesaikan Fubuki Puzzle yang berukuran 2x2, yang berarti hanya bisa diisi oleh angka 1, 2, 3, dan 4. Soal yang harus diselesaikan dapat dilihat di bawah ini.



Gambar 4. Contoh soal 2x2



Gambar 5. Pohon ruang status statis persoalan Fubuki Puzzle 2x2



Gambar 6. Pohon ruang status dinamis persoalan Fubuki Puzzle 2x2 yang dibentuk selama pencarian dengan algoritma backtracking

Sesuai dengan yang telah disebutkan di atas, hal pertama yang dilakukan adalah mengisi $x[1]$ dengan 1. Setelah itu, maka kita akan mencoba mengisi $x[2]$ dengan 2. Namun karena $x[1] + x[2]$ seharusnya berjumlah 4, bukan 3

(1+2), maka cabang tidak dibangkitkan lagi dan program melakukan runut balik.

Pada percobaan berikutnya, program mencoba mengisi $x[2]$ dengan 3. Karena hal ini memenuhi syarat penjumlahan untuk baris 1, maka program akan membangkitkan simpul 9 dan 11.

Program akan mencoba mengisi $x[3]$ dengan 2. Namun karena syarat penjumlahan kolom tidak dipenuhi, yaitu $1 + 2 = 3$, bukan 5 seperti yang diinginkan soal, maka program akan melakukan runut balik.

Program akan mengisi $x[3]$ dengan 4. Dan karena sampai langkah ini memenuhi syarat penjumlahan, maka $x[4]$ pasti diisi oleh 2.

III. KESIMPULAN

- a. Algoritma runut balik (Backtracking) dapat menyelesaikan permasalahan dalam Fubuki Puzzle
- b. Backtracking merupakan solusi efektif untuk permasalahan ini

REFERENSI

- [1] Horowitz, Ellis, & Sartaj Sahni, 1978, *Fundamental of Computer Algorithms*, Pitman Publishing Limited.
- [2] Munir, Rizaldi. 2005, strategi Algoritmik, Bandung.