Application of Greedy Algorithm to Check Out Wish List on Online Shopping

Haikal Lazuardi Fadil – 13519027
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
13519027@std.stei.itb.ac.id

***Abstract*—Online shopping has been a part of our everyday life that is now hard to separate with our ever-growing daily needs. However, excessive consumption may occur in uncontrolled shopping and useless stuffs were often bought carelessly. Using greedy algorithm, we could create a program that could control and sort things in our wish list or shopping cart on online shopping application.**

Keywords—greedy; algorithm, shopping, wishlist

I. Introduction

In human life, shopping has been a vital routine that can't be taken out easily since it is very essential to fulfill daily needs. Humans have been trading things from long ago and as humanity advanced, we now had money as our universal payments for goods.

With nowadays technology rapidly growing, shopping has taken a new form from the old way. Today, we don't have to walk to a store or a merchant and pick the thing we want to buy by ourselves before paying those with physical money. Now, all we need to do is open an application in our phone, choose the items that have taken our interest, and proceed to checkout. This development sure ease life in ways we could not imagine.

In a conventional store, we got a cart that we could fill with the things we would like to buy. It is a fact that that cart limited things we could check out since if we bought too much the cart would be brimmed with things and may spill over. Meanwhile, on online shopping application or website, we could simply add so many things to our chart to a bigger limit and even add them on wish list that is mostly holding almost infinite limit.

When it comes to payment, in conventional store we were mostly aware of the amount of money that we spend since some places requires us to pay with physical money. If we don't have enough money, we would most likely return some of the less important stuffs we were intended to buy. This would set us our limit in buying things. On the other hand, when presented to check out on online shop, our digital wallet was most likely easy to reach and to transfer some of its balance to meet the amount required. This enabled us to buy more expensive things in an easier way.

In some cases, it is a great help. Now we don't have to think twice about purchasing anything and the list of the things we want could be stored in a safe place. But this easiness promotes excessive consumption behavior that may be bad in the long run.

This kind of behavior may occur and be a weak spot for a lot of people, especially those who are still not yet reached maturity and don't have any stable income for a living. This could be a plague and a double-edged sword to all the improvement and the hope of future that lies before our eyes.

In this essay, writer would like to share their view regarding the use of greedy algorithm in checking out things on online shopping.

II. Theoritical Foundation

A. Online Shop

Online Shop is a form of electronic commerce that allows consumers to directly purchase goods or services from a seller over the Internet using a web browser or a mobile app. Consumers find a product of interest by visiting the website of the retailer directly or by searching among alternative vendors using a shopping search engine, which displays the same product's availability and pricing at different e-retailers. As of 2020, customers can shop online using a range of different computers and devices, including desktop computers, laptops, tablet computers and smartphones.

An online shop evokes the physical analogy of buying products or services at a regular "bricks-and-mortar" retailer or shopping centre; the process is called business-to-consumer online shopping. When an online store is set up to enable businesses to buy from another businesses, the process is called business-to-business online shopping. A typical online store enables the customer to browse the firm's range of products and services, view pictures or images of the products, along with information about the product specifications, features and prices.

Figure 1. Online shopping illustration

Source: https://cerebra.org.uk/get-involved/give-as-you-shop/

Online stores usually enable customers to look up for features to find specific models, brands or items. Online customers must have access to the Internet and a valid method of payment in order to complete a transaction, such as a credit card, debit card, or a service such as PayPal. For physical products (e.g., paperback books or clothes), the e-tailer ships the products to the customer; for digital products, such as digital audio files of songs or software, the e-tailer usually sends the file to the customer over the Internet. The largest of these online retailing corporations are Alibaba, Amazon.com, and eBay.

Simple shopping cart systems allow the off-line administration of products and categories. The shop is then generated as HTML files and graphics that can be uploaded to a webspace. The systems do not use an online database. A high-end solution can be bought or rented as a stand-alone program or as an addition to an enterprise resource planning program. It is usually installed on the company's web server and may integrate into the existing supply chain so that ordering, payment, delivery, accounting and warehousing can be automated to a large extent. Other solutions allow the user to register and create an online shop on a portal that hosts multiple shops simultaneously from one back office.

B. *Greedy Algorithm*

A greedy algorithm is any algorithm that follows the problem-solving heuristic of making the locally optimal choice at each stage. In a lot of problems, a greedy strategy does not usually return an optimal solution, but nonetheless, a greedy heuristic may yield locally optimal solutions that approximate a globally optimal solution in a reasonable amount of time.

For example, a greedy strategy for the travelling salesman problem (which is of a high computational complexity) is the following heuristic: "At each step of the journey, visit the nearest unvisited city." This heuristic does not intend to find a best solution, but it terminates in a reasonable number of steps; finding an optimal solution to such a complex problem typically requires unreasonably many steps. In mathematical optimization, greedy algorithms optimally solve combinatorial problems having the properties of matroids, and give constant-factor approximations to optimization problems with submodular structure.

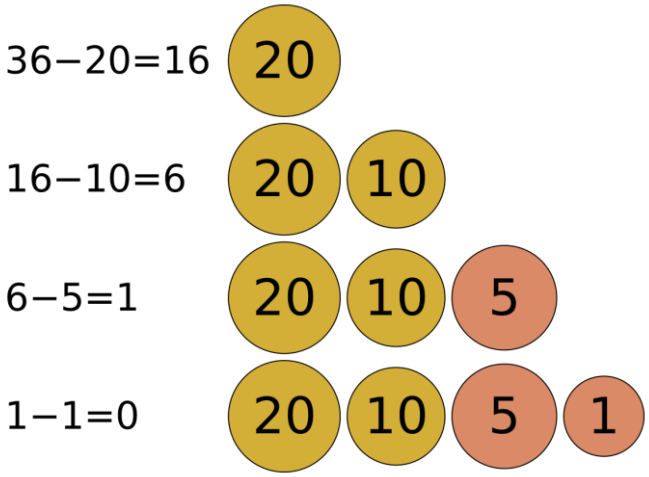

Figure 2. Greedy Algorithm Illustration

Source: https://cerebra.org.uk/get-involved/give-as-you-shop/

In general, greedy algorithms have five components:

1. A candidate set, from which a solution is created
2. A selection function, which chooses the best candidate to be added to the solution
3. A feasibility function, that is used to determine if a candidate can be used to contribute to a solution
4. An objective function, which assigns a value to a solution, or a partial solution, and
5. A solution function, which will indicate when we have discovered a complete solution

Greedy algorithms produce good solutions on some mathematical problems, but not on others. Most problems for which they work will have two properties:

We can make whatever choice seems best at the moment and then solve the subproblems that arise later. The choice made by a greedy algorithm may depend on choices made so far, but not on future choices or all the solutions to the subproblem. It iteratively makes one greedy choice after another, reducing each given problem into a smaller one. In other words, a greedy algorithm never reconsiders its choices. This is the main difference from dynamic programming, which is exhaustive and is guaranteed to find the solution. After every stage, dynamic programming makes decisions based on all the decisions made in the previous stage, and may reconsider the previous stage's algorithmic path to solution.

C. *Knapsack Problem*

The knapsack problem is a problem in combinatorial optimization: Given a set of items, each with a weight and a value, determine the number of each item to include in a collection so that the total weight is less than or equal to a given limit and the total value is as large as possible. It derives its name from the problem faced by someone who is constrained by a fixed-size knapsack and must fill it with the most valuable items. The problem often arises in resource allocation where the decision makers have to choose from a set of non-divisible projects or tasks under a fixed budget or time constraint, respectively.

Figure 3. Knapsack Problem

Source: https://www.researchgate.net/figure/0-1-knapsack-problem_fig1_327326005

Knapsack problems appear in real-world decision-making processes in a wide variety of fields, such as finding the least wasteful way to cut raw materials, selection of investments and portfolios, selection of assets for asset-backed securitization, and generating keys for the Merkle–Hellman and other knapsack cryptosystems.

III. Implementation

Often time, we have things that are not quite important in our online shopping cart and they are expensive. However, in online shopping, it is not rare for us to check them all out at once and pay a large amount of money because there is so little limitation of the things we could buy.

Using greedy algorithm's knapsack problem, we could set some limit for the thing we would like to check out so we can sort out unnecessary things and use our money more wisely. Further explanation would be described as below:

A. *Application*

In checking things out from our shopping cart there are few things that we need to pay attention to other than the price. Each thing has its own usefulness to our current condition at a certain time. If one person just checks everything in their shopping cart out without considering each item's necessity for his current condition, it will cost him a big time in the long run. That kind of things could easily be avoided using computer science way of thinking.

In this case, greedy algorithm is used with the maximum optimalisation with the hope that the items checked out fulfilled the current need and do not spend more money than the limit set beforehand.

Greedy Algorithm Element:

- Candidate Set:

  Collection of all things exist in the wish list of an online shopping with price under the limit set that could be checked out.

- Solution Set:

  Collection of the things with highest usefulness for the current condition that could be collected with the total of price no more than the price limit set

- Selection Function:

  This is the function that do the checking for total of the film so they do not exceed the price limit set.

- Objective Function:

  The function that makes sure that all solution set have met the constraint and is consisting of the items with highest usefulness.

B. *Study Case*

Suppose there is a person who was addicted to shopping and would like to set some limit to his shopping habit so that he would spend no more than $70 for each check outs rather than checking everything out as he used to do.

He also made a list of each item's usefulness for his current condition in his wish list so that he wouldn't check out the things that are unnecessary for him.

This kind of problem can be solved by imagining this problem as integer knapsack problem.

Items in the wish list:

Figure 4. Crystal Necklace

Source: https://www.amazon,com

Figure 5. Pillow

Source: https://www.amazon.com

Figure 6. Waffle maker

Source: https://www.amazon.com

Figure 7. Gardening set

Source: https://www.amazon.com

Figure 8. Brigid Statue

Source: https://www.amazon.com

Figure 9. Cable

Source: https://www.amazon.com

Figure 10. Binocular

Source: https://www.amazon.com

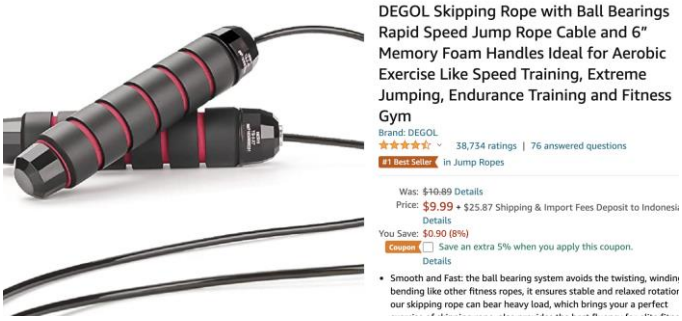

Figure 11. Skipping Rope

Source: https://www.amazon.com

Price of each item listed in the above will be presented in a table along with their usefulness scale rated by this person to match his current condition and necessity.

No	Item	Usefulness	Price
1	Crystal Necklace	3	$14.99
2	Pillow	6	$8.99
3	Waffle Maker	6	$9.99
4	Gardening Set	7	$33.99
5	Brigid Statue	2	$83.25
6	Cable	8	$14.02
7	Binocular	5	$39.94
8	Skipping Rope	7	$9.99

C. Experiment

There are three algorithm approach that could be applicated in the search of the set of items the person should check out. These closure are derived from knapsack problem that are related to price, usefulness, and the combination of both.

Greedy by price

- For each step, the cheapest item is selected
- Maximation could be done by purchasing as many item as possible within the set limit

Item	Usefulness	Price	Solution
Crystal Necklace	3	$14.99	1
Pillow	6	$8.99	1
Waffle Maker	6	$9.99	1
Gardening Set	7	$33.99	0
Brigid Statue	2	$83.25	0
Cable	8	$14.02	1
Binocular	5	$39.94	0
Skipping Rope	7	$9.99	1
		Total Price	$57.98
		Total Usefulness	30

Greedy by Usefulness

- For each step, the item chosen is the item with the highest usefulness
- Profit maximation is done by purchasing the most useful things

Item	Usefulness	Price	Solution
Crystal Necklace	3	$14.99	0
Pillow	6	$8.99	1
Waffle Maker	6	$9.99	0
Gardening Set	7	$33.99	1
Brigid Statue	2	$83.25	0
Cable	8	$14.02	1
Binocular	5	$39.94	0
Skipping Rope	7	$9.99	1
		Total Price	$67.99
		Total Usefulness	28

Greedy by Density

- For each step, the item's usefulness will be divided by the price
- Profit maximation is done by purchasing the items with biggest comparison of usefulness per price.

Item	Usefulness	Price	Usefulness/Price	Solution
Crystal Necklace	3	$14.99	0.2	1
Pillow	6	$8.99	0.66	1
Waffle Maker	6	$9.99	0.6	1
Gardening Set	7	$33.99	0.2	0
Brigid Statue	2	$83.25	0.024	0
Cable	8	$14.02	0.57	1
Binocular	5	$39.94	0.125	0
Skipping Rope	7	$9.99	0.7	1
Total Price				$57.98
Total Usefulness				28

Comparison of the three method would be best described in the table below:

Item	Usefulness	Price	Solution		
			1	2	3
Crystal Necklace	3	$14.99	1	0	1
Pillow	6	$8.99	1	1	1

Waffle Maker	6	$9.99	1	0	1	
Gardening Set	7	$33.99	0	1	0	
Brigid Statue	2	$83.25	0	0	0	
Cable	8	$14.02	1	1	1	
Binocular	5	$39.94	0	0	0	
Skipping Rope	7	$9.99	1	1	1	
Total Price			$57.98	$67.99	$57.98	$67.99
Total Usefulness			30	28	30	28

From the table below it is shown that the result earned from the three methods gave different values. The closure that pinpoint on price and density returned crystal necklace, pillow, waffle maker, cable, and skipping rope while the one that focused on the usefulness got pillow, gardening set, cable, and skipping rope.

IV. Conclusion

Within this essay, writer has discussed thoroughly about the application of greedy algorithm in the practice of human daily life which is online shopping. According to the experiment and the research conducted by the writer, it turns out greedy algorithm is quite handy to solve simple life problem like this.

The algorithm to decide which items we should checkout from our wish list is just a simple example of many algorithm that could solve this problem. However, writer chose to address this one because the linearity congruence just shows that

Video Link at Youtube

https://youtu.be/KYft29RFKHU

Acknowledgment

Writer is grateful to God for completing this paper on time. Writer also thanks the IF2211 Strategi Algoritma lecturers for the services in teaching this course for one semester. Last but not least, writer also thank all references who have contributed to the production of this paper.

References

informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2019-2020/Algoritma-Greedy-(2020).pdf accessed on Tuesday May 11, 2021 00.00 AM

https://www.geeksforgeeks.org/0-1-knapsack-problem-dp-10/ accessed on Tuesday May 11, 2021 00.00 AM

[1] S. Martello, P. Toth, Knapsack Problems: Algorithms and Computer Implementations, John Wiley and Sons, 1990

[2] Horowitz, Ellis; Sahni, Sartaj (1974), "Computing partitions with applications to the knapsack problem", *Journal of the Association for Computing Machinery*,

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, 26 April 2021

Haikal Lazuardi Fadil - 13519027