UNIVERSIDADE FEDERAL DOS VALES JEQUITINHONHA E MUCURI – UFVJM

DISCIPLINA: AEDS III

PROFESSOR: MARCELO FERREIRA REGO

ALUNO: MARLON PARANHOS DUARTE

ATIVIDADE: KNAPSACK PROBLEM (PROBLEMA DA MOCHILA)

INTRODUÇÃO

Proposta

Implementar 3 diferentes algoritmos utilizando abordagens gulosas para resolver o Problema da Mochila 0/1 (Knapsack Problem) respeitando uma dada capacidade P.

Situação

- Um ladrão acha *n* itens numa loja;
- Item *i* vale v_i unidades (dinheiro, e.g., R$, US$, etc);
- Item *i* pesa p_i unidades (kg, etc);
- v_i e p_i são inteiros;
- Consegue carregar *P* unidades no máximo;
- Deseja carregar a "carga" mais valiosa.

IMPLEMENTAÇÃO

Foram implementados 3 algoritmos para 3 possíveis soluções do problema, são eles:

Carga mais valiosa:

A programação dinâmica nos ajuda a otimizar os problemas sem verificar todos os casos possíveis, podendo assim chegar a uma solução ótima. Para o algoritmo mochila, o que queremos é preenchê-la com os elementos de maior valor e de menor peso.

- A função ***cargaOtima()*** recebe como parâmetro os dados sobre a capacidade da mochila, pesos, benefícios e número de itens disponíveis que poderão ser colocados dentro dela;
- É criada uma tabela de n+1 linhas e n+1 colunas para que as verificações sejam feitas, onde a primeira linha e coluna são preenchidas com zero e *n* representa a quantidade de elementos a serem inseridos na mochila;
- É então executado um laço de repetição, que é o responsável por fazer todos os cálculos e verificações necessárias para que o peso total dos itens não exceda a capacidade da mochila;
- Por fim, a função retorna o valor total do benefício dos elementos inseridos dentro da mochila.

Pelos melhores valores:

O cálculo feito pelos melhores valores (ou benefícios) foi implementado da seguinte maneira:

- A função ***melhorValor()*** recebe como parâmetro os dados sobre a capacidade da mochila, pesos, benefícios e número de itens disponíveis que poderão ser colocados dentro dela;
- Como os valores podem ser inseridos na forma crescente, decrescente ou aleatória, foi feita uma função para que seja possível ordenar os vetores na ordem crescente onde, neste caso, sua leitura é feita de modo inverso, ou seja, do maior para o menor valor;
- É então executado um laço de repetição, que é o responsável por fazer todos os cálculos e verificações necessárias para que o peso total dos itens não exceda a capacidade da mochila;
- Por fim, a função retorna o valor total do benefício dos elementos inseridos dentro da mochila.

Pelos menores pesos:

O cálculo feito pelos melhores valores (ou benefícios) foi implementado da seguinte maneira:

- A função ***menorPeso()*** recebe como parâmetro os dados sobre a capacidade da mochila, pesos, benefícios e número de itens disponíveis que poderão ser colocados dentro dela;
- Como os valores podem ser inseridos na forma crescente, decrescente ou aleatória, foi feita uma função para que seja possível ordenar os vetores na ordem crescente onde, para assim já selecionar os elementos de menor peso;
- É então executado um laço de repetição, que é o responsável por fazer todos os cálculos e verificações necessárias para que o peso total dos itens não exceda a capacidade da mochila;
- Por fim, a função retorna o valor total do benefício dos elementos inseridos dentro da mochila.

ANÁLISE DE COMPLEXIDADE

Resolver este tipo de problema utilizando força bruta se torna extremamente inviável, pois teríamos que fazer todas as combinações possíveis para poder saber qual seria a melhor para maximizar o valor dos elementos dentro da mochila.

A complexidade para este caso seria de *O(2^n)*, onde *n* representa o número de elementos, provando a inviabilidade do método pelo fato de seu custo ser exponencial de acordo com a variação da quantidade de elementos.

ANÁLISE DOS RESULTADOS

Os resultados a serem apresentados foram baseados nas instâncias fornecidas pelo professor.

(1) - Carga mais valiosa: Resultados obtidos a partir da escolha ótima para cada situação (instância) utilizando programação dinâmica para a solução.

(2) - Melhores valores: Resultados obtidos a partir da soma dos elementos com melhor benefício até que a capacidade da mochila seja excedida.

(3) - Menores pesos: Resultados obtidos a partir da soma dos elementos de menor peso até que a capacidade da mochila seja excedida.

Instância	Itens	Capacidade	Método (1)	Método (2)	Método (3)
1	10	269	295	288	74
2	20	878	1034	1026	677
3	4	20	35	28	33
4	4	11	23	13	16
5	15	375	DOUBLE	DOUBLE	DOUBLE
6	10	60	52	100	2
7	7	50	107	166	12
8	23	10000	9767	10735	7604
9	5	80	130	130	105
10	20	879	1025	982	750

CONCLUSÃO

De modo geral, não se trata de um problema extremamente complexo, pois não exige conhecimento extenso a respeito de programação dinâmica e algoritmos gulosos. A maior dificuldade foi elaborar a proposta de solução para o método de carga ótima, onde foi necessário buscar ajuda em fontes externas.

Inicialmente, implementei a função para encontrar a carga ótima utilizando ponteiros, mas obtive alguns erros ao manipular o vetor com os elementos, pois desejava retornar o número de itens ali contidos para poder facilitar na hora de executar os cálculos, e foi onde resolvi passar o número de elementos como parâmetro.

Pela falta de prática com a linguagem C++, tive também dificuldades para conseguir ordenar os itens dos vetores (do menor para o maior), e quando consegui fazer, lembrei que existia uma função nativa da linguagem que resolvia este problema, a ***qsort()***.

REFERÊNCIAS

<https://stackoverflow.com/questions/4108313/how-do-i-find-the-length-of-an-array> How do I find the length of an array? – Acessado em 24/05/2018

<http://www.cplusplus.com/reference/cstdlib/qsort/> Sort elements of array – Acessado em 25/05/2018

<https://youtu.be/F-wu-wkdv-M> Programação dinâmica - o problema da mochila (2° exercício resolvido) – Acessado em 25/05/2018

<http://www.codcad.com/lesson/39> O Problema da Mochila – Acessado em 25/05/2018

<http://xcodigoinformatico.blogspot.com.br/2012/09/algoritmo-de-la-mochila-01-con.html> Algoritmo de la mochila 0/1 con Programación Dinámica – Acessado em 24/05/2018