

Trabalho 3 - Análise de Algoritmos

Entrega: 1 de Julho

Esse trabalho consiste em implementar um algoritmo de programação dinâmica para uma variação do problema da mochila visto em sala chamado **10-Knapsack**.

O trabalho deve ser realizado **em dupla**, e pode ser feito em qualquer linguagem.

O que tem que ser entregue **por email, em um zip com os nomes dos membros da dupla**:

1. Relatório de 2-3 páginas com o conteúdo pedido abaixo
2. Código fonte
3. Executável **windows** (não precisa caso feito em Python)

Note que utilizarei sistema para controle de plágio de código, como o MOSS.¹

Tarefa 1: Desenho do algoritmo

O problema **10-Knapsack** é o seguinte: temos uma mochila de tamanho B e n itens, onde o item i tem tamanho w_i e valor v_i . Porém agora temos **10 unidades de cada item**. O objetivo é selecionar uma quantidade de cada itens (de 0 a 10) de forma que caibam na mochila (i.e., soma dos tamanho é menor que B) e que obtenha valor máximo.

Assuma que todos os dados $B, w_1, \dots, w_n, v_1, \dots, v_n$ são inteiros não-negativos.

Ex: Considere a instancia com $B = 8$, $n = 2$, $(w_1, w_2) = (3, 2)$, e $(v_1, v_2) = (2, 1)$. Nesse caso a solução ótima é selecionar 2 unidades do elemento 1 e uma unidade do elemento 2, obtendo valor 5.

Ex 2: Considere a instancia com $B = 22$, $n = 2$, $(w_1, w_2) = (2, 1)$, e $(v_1, v_2) = (5, 1)$. Nesse caso a solução ótima é selecionar 10 unidades do elemento 1 e duas unidades do elemento 2, obtendo valor 52.

A Tarefa 1 é a seguinte: Dada uma instancia, seja $OPT(i, b)$ o valor da solução ótima quando apenas consideramos os itens de 1 a i e quando o valor da mochila é b . Encontre uma equação de recorrência para $OPT(i, b)$.

Note que você **não pode simplesmente construir uma nova instancia que replica cada item 10 vezes e utilizar a solução para o problema da mochila visto em aula**, você tem que dar uma equação de recorrência para $OPT(i, b)$ exatamente como definido acima.

No relatório você deve reportar: **A equação de recorrência encontrada**

¹<https://theory.stanford.edu/~aiken/moss>

Tarefa 2: Implementação do algoritmo

Na segunda tarefa você deve implementar o algoritmo de programação dinâmica para resolver o problema 10-Knapsack baseado na equação de recorrência encontrada na Tarefa 1. Note que o algoritmo precisa retornar **a quantidade de cada item que deve ser selecionada**, e **não só o valor** da solução ótima.

Para validar seu algoritmo, você deve executá-lo nas 8 instâncias providas (encontradas na página do curso).

No relatório você deve reportar:

1. O código principal do seu algoritmo, ou seja, aquele que preenche a tabela de memoização (seu algoritmo pode ser iterativo ou recursivo).
2. A solução ótima (e não só o seu valor) para as 8 instancias providas.