

Aluno: André Santos Rocha

Ra: 235887

Problema 1: A - The Knapsack Problem

O problema consiste no problema da mochila. Dada uma mochila de capacidade S ($1 \leq S \leq 2000$), podemos colocar N itens dentro da mochila ($1 \leq N \leq 2000$). Cada item possui um valor c e um peso p ; devemos posicionar os itens na mochila de modo a maximizar o valor total dos itens na mochila, respeitando a capacidade de peso da mochila.

1.1 Ideia de solução

Para esse problema, utilizamos programação dinâmica. Considere o seguinte raciocínio: dado uma entrada de itens, o item n pode ou não fazer parte da solução. Se n fizer parte da solução, teremos que nossa resposta será o valor C_n de n somado à melhor solução do problema da mochila com capacidade $S - p$, considerando os primeiros $n-1$ itens avaliados. Caso n não faça parte da solução, a resposta será dada pela solução do problema da mochila com capacidade S , considerando os $n-1$ primeiros itens avaliados. Faremos uma matriz de dp , onde $dp[k][d]$ é a resposta do problema da mochila considerando uma mochila de capacidade d com os k primeiros itens. Daí computamos o restante da matriz da seguinte forma:

- $z[0][d] = 0$
- $z[k][0] = 0$
- $z[k][d] =$
 - $z[k-1][d]$, se $P_k > d$
 - $\max\{z[k-1][d], z[k-1][d-P_k] + C_k\}$, se $P_k \leq d$

Problema 2: C - Kefa and First Steps

O problema consiste em determinar o comprimento da maior subsequência não-decrescente, dada sequência de n inteiros.

1.1 Ideia de solução

Para este problema, percorremos a sequência de entrada até o penúltimo elemento. Durante essa iteração, verificamos, para cada elemento na posição i , verificamos se o elemento na posição $i+1$ é maior ou igual. Caso seja, adicionamos 1 na nossa variável len que guarda o comprimento da subsequência. Caso contrário, setamos o valor de len para 1. Para garantir que teremos a maior subsequência, é preciso também armazenar em uma variável max o maior comprimento obtido até agora. A cada iteração se len for maior que max , então max recebe o valor de len . Dessa forma garantimos que sempre estaremos

computando o len de uma subsequência não-decrescente e que essa subsequência é a maior possível.