

Algorithm Description

Primeiro é colocado num array de eventos(uma estrutura criada para este problema), todos os eventos e as suas características.

Após todos os dados estarem inseridos, é usado um quicksort para ordenar os eventos em relação à sua deadline.

Guardamos também o evento com a maior deadline para usar no algoritmo que dará a melhor combinação de eventos.

Depois disto, é inicializada uma tabela, em que a primeira coluna e a primeira linha têm o valor 0.

Começamos por iterar pelo numero de eventos, variável i , e por cada evento corremos um ciclo desde 1 até ao maior deadline recebido, variável j .

Primeiro verificamos se conseguimos usar o evento atual, e o critério é se a sua duração é inferior ou ao seu deadline ou ao deadline máximo que estamos a avaliar (variável j), o que for menor destes dois.

Se a duração do evento não for inferior a este valor, então não podemos usar este evento, e para este i e j na matriz soma, guardamos o valor de usar o evento anterior (ou seja, $DP[i][j] = DP[i-1][j]$).

No entanto, se o conseguirmos usar, vamos guardar na matriz soma o melhor resultado (lucro) entre usar este evento e não o usar. Para usar este evento, temos que ir buscar o valor da matriz soma que corresponde ao evento anterior ($i-1$) e o deadline resultante (o menor entre j e o do evento menos a duração do evento que usámos) e somar ao lucro associado com o evento.

Data Structures

Para realizar este problema, criamos uma estrutura especifica denominada evento, esta estrutura possui 3 int, um deadline, um duration, e um profit.

Já depois de inserido os dados, criamos um array de eventos, cada um com as características acima enumeradas.

Criámos também uma matriz DP, que guarda os valores de subproblemas já calculados.

Correctness

Devido a termos obtido a classificação de 150 pontos iremos explicar porque a nossa abordagem é a correta.

A abordagem que decidimos seguir é a correta, pois apenas adaptamos o pseudocodigo dado pelo professor, estando este fundamentalmente correto podemos concluir que as nossas alterações mínimas não alteraram em nada a validade do código.

Também pudemos verificar a validade da nossa abordagem com os testes oferecidos pelo professor.

Algorithm Analysis

A complexidade deste algoritmo é $O(nW)$, em que “W” é a deadline máxima e “n” é o número de eventos que irão ser inseridos na matriz.

References

A única referência que usamos foi a do pseudocódigo oferecido pelo professor nos slides teóricos.

Team members that implemented the approach

Student ID : 2016230975 Name: João Miguel Rainho Mendes

Student ID : 2016228672 Name: João Filipe Sequeira Montenegro

Student ID : 2016240139 Name: Flávio Amaral Fernandes