Trabalho 1 - Algoritmos e Estruturas de Dados II

Prof. Dr. João do E. S. Batista Neto

Março 2018

1 Introdução

Esse trabalho é composto por duas partes. Ambas devem ser resolvidas utilizando-se os algoritmos vistos em aula. Para testar sua solução, utilize os casos de exemplo disponibilizados ao fim da descrição de cada parte e desenvolva seus próprios casos de teste.

2 Parte 1 - Ordenaçao Topológica

Escreva um programa que encontre e imprima uma ordenação topológica de um grafo direcionado. Caso exista mais de uma ordenação possível, encontre a menor delas em ordem lexicográfica.

O grafo é composto por n vértices numerados de 1 a n. É garantido que o grafo descrito é acíclico (mas não necessariamente conexo), portanto sempre existirá ao menos uma ordenação topológica válida. Além disso, não existem laços nem arestas múltiplas.

A primeira linha da entrada consiste de dois inteiros separados por espaço, o número de vértices n e o número de arestas m.

Cada uma das m linhas seguintes é composta por dois inteiros, u e v, separados por espaços, descrevendo uma aresta direcionada de u para v.

Imprima na saída padrão n números distintos de 1 a n que descrevam a menor (lexicograficamente) ordenação topológica do grafo dado.

Os limites da entrada são descritos a seguir:

- $1 \leq n \leq 10^5$
- $0 \leq m \leq min\left(10^5, \frac{n(n-1)}{2}\right)$
- $1 \leq u, v \leq n$, $u \neq v$, para cada uma das arestas.

Exemplo de entrada:

```
5 3
1 2
1 3
5 4
```

Exemplo de saída:

```
1 2 3 5 4
```

Note que "5 4 1 2 3", "1 3 2 5 4" e "1 2 5 4 3" são possíveis ordenações topológicas do grafo dado, mas nenhuma delas é a menor lexicograficamente. Em contrapartida, "1 2 3 4 5" é a menor sequência lexicográfica de 5 números distintos, porém não é uma ordenação topológica válida pois o vértice 4 vem antes do vértice 5 e existe uma aresta de 5 para 4.

3 Parte 2 - Menor Custo

Em uma pilha existem n objetos com diferentes volumes. É necessário guardá-los em k caixas para transporte e cada uma delas possui volume V. Os objetos são guardados segundo o seguinte algoritmo:

Para cada caixa de 1 a k, se o objeto no topo da pilha não cabe na caixa atual, essa caixa é lacrada e enviada para transporte. Caso contrário, há duas possibilidades: 1) retira-se o objeto do topo da pilha e esse é guardado na caixa atual e, em seguida, o processo continua para o próximo objeto na pilha; ou 2) a caixa atual é lacrada e não se remove o objeto do topo da pilha, continuando o processo a partir da próxima caixa.

Note que o algoritmo acima pode guardar os n objetos em k caixas de diversas maneiras, dependendo de qual opção foi feita na hora de guardar cada objeto. Além disso, dada uma maneira de se guardar esses objetos, é possível calcular o *custo* dessa organização da seguinte maneira:

Seja x_i o volume total dos objetos guardados na i-ésima caixa, para $i = 1, 2, ..., k$. O custo C é dado por:

$$C = \sum_{i=1}^{k} (V - x_i)^2$$

Ou seja, a soma do quadrado do volume restante em cada caixa.

Escreva um programa que, dado n, k, V e os pesos dos objetos $w_1, w_2, ..., w_n$, calcule qual o custo mínimo para se guardar os n objetos.

A primeira linha da entrada consiste de três inteiros separados por espaço, n, k e V.

A segunda e última linha da entrada é composta por n inteiros separados por espaço, $w_1, w_2, ..., w_n$.

Caso não seja possível guardar os n objetos em k caixas, imprima "-1" na saída padrão. Caso contrário, imprima o menor custo C possível de se obter seguindo os passos do algoritmo descrito.

Os limites da entrada são descritos a seguir:

- $1 \leq k \leq n \leq 100$
- $1 \leq V \leq 1000$
- $1 \leq w_i \leq V$, para $i = 1, 2, ..., n$

Exemplo de entrada:

```
2 2 10
5 5
```

Exemplo de saída:

```
50
```

Note que também seria possível colocar os dois objetos na primeira caixa, deixando a segunda vazia. Mas dessa forma teríamos um custo superior a 50:

$$C = (0)^2 + (10)^2 = 100 > 50$$