

UNIVERSIDADE FEDERAL DE SÃO JOÃO DEL REI
DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO
ALGORITMOS E ESTRUTURAS DE DADOS III

2º semestre de 2019

Professor: Leonardo Chaves Dutra da Rocha

Trabalho Prático 2

Data de Entrega: 10 de Outubro 2019.

Introdução

Zorc, o maior guerreiro do Mundo de Zambis, está recrutando seu exército para a grande batalha final contra o exército do Mundo de Xulambis. Ele saiu caminhando pelo seu mundo atrás de guerreiros de diferentes povos de seu mundo. A batalha ocorrerá no Mundo Xulambis e, portanto, Zorc transportará seu exército por meio de uma nave que suporta um peso máximo de W quilogramas. Para cada povo i que Zorc visita, os soldados possuem um peso w_i (supondo que todos os soldados do mesmo povo possuem o mesmo peso) e uma habilidade de guerra de h_i (supondo que todos os soldados do mesmo povo possuem a mesma habilidade).

O recrutamento dos soldados deve ser feita caminhando pelo Mundo de Zambis. Existem P povos diferentes e cada povo p_i tem soldados de peso w_i e habilidade h_i em quantidade ilimitada. Zorc é livre para recrutar quantos soldados quiser, desde que eles caibam na nave de transporte. Como o terreno do Mundo de Zambis é acidentado, não é possível ir de qualquer povo para qualquer povo. De fato, para cada povo p_i há um conjunto de povos $P(p_i)$ que podem ser alcançados a partir dele. Tão pouco é possível recrutar soldados para todo sempre. Cada passagem de uma povo i para uma povo j requer que se percorra uma distância d_{ij} metros e o como o momento da guerra se aproxima, Zorc não pode andar por mais que D metros no total.

Você precisa auxiliar Zorc a encontrar o caminho que maximiza a habilidade total dos soldados que serão recrutados! Para este trabalho, você deverá apresentar uma solução que utilize o paradigma de programação dinâmica e uma solução que utiliza alguma heurística.

1 Entrada

A entrada será um arquivo no seguinte formato: na primeira linha estará um número inteiro K que apresenta quantas instâncias o arquivo possui. Na próxima linha começa a primeira instância.

Na primeira linha de cada instância contém o número de povos P , a distância D , em metros, que Zorc pode andar, o peso W da nave em gramas e a quantidade C de caminhos que existem entre as povos. A seguir, nas próximas P linhas estarão descritos o povo p_i , o peso w_i e a habilidade h_i de seus soldados. A habilidade H dos soldados pode variar de 1(pior qualidade) até 10(melhor qualidade). Nas próximas C linhas está descrito o povo p_i e p_j e a distância d_{ij} entre eles. Vale lembrar que, caso nessas C linhas não seja colocado o povo p_x e p_y e sua distância d_{xy} , significa que não é possível alcançar p_y a partir de p_x e vice-versa Na próxima linha começa a próxima instância.

Os valores P, D, C, H e W serão representados por um inteiro.

Veja o exemplo a seguir:

```
2
6 10 310 7
1 70 2
2 100 3
3 20 7
4 90 4
5 20 3
6 10 1
```

```

1 2 3
1 5 2
2 3 4
2 4 2
3 6 3
4 5 3
4 6 5
5 6 3 17 6
1 10 10
2 30 9
3 1 1
4 3 6
5 2 4
1 2 3
1 3 2
1 4 2
2 3 1
3 5 3
4 5 4

```

Saída

A saída deve mostrar o caminho que foi percorrido para conseguir maximizar a habilidade dos soldados recrutados. Para cada caminho, imprima a habilidade total da exército seguida pelas tuplas: povo visitado e a habilidade dos soldados recrutados no povo.

No primeiro exemplo abaixo, a visita começa no povo 4 onde nenhuma soldado é recrutado, segue para a povo 6 onde é recrutado apenas um soldado e termina no povo 3 onde 15 soldados são recrutados. A habilidade total do exército é 106.

Note que pode haver mais de um caminho que maximize a habilidade dos soldados recrutados. Não existe critério de desempate: escolha qualquer um dos caminhos que tenha a habilidade máxima.

Veja a saída para o exemplo anterior:

```

106 4 0 6 1 3 15
634 4 105 5 1

```

Documentação

A documentação não deve exceder 12 páginas e deve conter pelo menos os seguintes itens:

- Uma **introdução** do problema em questão.
- **Modelagem e solução** proposta para o problema. O algoritmo deve ser explicado de forma clara, possivelmente através de pseudo-código e esquemas ilustrativos. **Lembre-se de mostrar que o problema pode ser resolvido utilizando programação dinâmica**, isto é, que ele possui subestrutura ótima e sobreposição de problemas.
- **Análise de complexidade** de tempo e espaço da solução implementada.
- **Experimentos e análise de resultados** variando-se o tamanho da entrada e quaisquer outros parâmetros que afetem significativamente a execução. Explique claramente o que significa cada resultado alcançado
- Especificação da(s) **máquina(s) utilizada(s)** nos experimentos realizados.

- Uma breve **conclusão** do trabalho implementado.

Observações:

- O código fonte do trabalho deve ser submetido para compilação e execução em ambiente Linux, tendo como padrão os computadores dos laboratórios do DCOMP.
- Deve ser escrito na linguagem C (trabalhos implementados em outras linguagens como C++/Java/Python e outras não serão aceitos);
- As estruturas de dados devem ser alocadas dinamicamente e o código deve ser modularizado utilizando os arquivos .c .h.
- O utilitário Make deve ser utilizado para compilar o programa;
- A saída deve ser impressa no arquivo pedido seguindo estritamente o formato da especificação caso contrário o resultado será considerado errado;
- O arquivo executável deve ser chamado de **tp2** e deve receber como parâmetro (-i) apenas o nome do arquivo de entrada de dados e (-o) para o arquivo de saída. Não serão aceitos outros nomes de executáveis além dos mencionados.
- Faça seu código de forma legível

Avaliação

Deverão ser entregues:

- listagem das rotinas;
- documentação contendo:
 - descrição das soluções e estruturas de dados utilizadas;
 - análise da complexidade das rotinas;
 - análise dos resultados obtidos.

Distribuição dos pontos:

- execução
 - execução correta: 60%
 - saída legível: 20%
- estilo de programação
 - código bem estruturado: 20%
- documentação
 - comentários explicativos: 30%
 - análise de complexidade: 20%
 - análise de resultados: 50%

A nota final é calculada como a média harmônica entre execução (E) e documentação (D):

$$\frac{D * E}{\frac{D+E}{2}}$$