Problemas sobre grafos nas competições

Nas maratonas da SBC cerca de 2 problemas, em 12 envolvem grafos

Os problemas não são problemas de pesquisa, mas é necessário modelar bem e, muitas vezes, adaptar algoritmos conhecidos.

Problemas, às vezes envolvendo milhões de dados têm que ser resolvidos em 1 segundo, em geral, o que força a busca dos mais eficientes algoritmos. (1 seg $\cong 10^8$ instruções)

A escolha do algoritmo é feita observando-se o tamanho da entrada:

Até n = 1.000 pode ser possível usar algoritmo $O(n^2)$

Até n = 100.000 normalmente deve-se usar $O(n \log n)$

Acima de 100.000 normalmente deve-se usar $O(n)$

Problemas sobre grafos nas competições

90% dos alunos usam C++, devido à STL, que tem prontos os tratamentos para as diversas estrutuas de dados

Em C++ existe uma estrutura de dados especial: VECTOR que é um misto de vetor e lista encadeada. Normalmente é a estrutura preferida pelos alunos.

VECTOR são vetores cujo tamanho é automaticamente estendido quando necessário. É como se fossem listas encadeadas que possibilitam acesso por índice em qualquer parte da lista.

VECTOR é, sobretudo, uma estrutura de dados prática, sem estudos teóricos associados.

Problemas em Grafos em Maratonas da SBC

(disponíveis no site do URI)

1931 - Mania de Par
2666 - Imposto Real
2962 - Arte Valiosa
1442 - Desvio de Rua
1391 - Quase o Menor Caminho
2882 - Gasolina
1476 - Caminhão
1490 - Torres que Atacam

1931 - Mania de Par

Contexto: Patricia vai fazer uma viagem onde todas as estradas são bidirecionais e têm sempre um pedágio em cada trecho. Dado o mapa das estradas quer-se saber qual o pedágio mínimo que ela vai pagar, com a restrição de que tem que ser um número par de pedágios.

Entrada: Um caso de teste. Na primeira linha, N e M ($2 \le N \le 10^4$, $0 \le M \le 50000$). A seguir vêm M linhas, com 3 inteiros C_1, C_2, indicando o par de cidades ligados e G ($\le 10^4$) o pedágio. Patrícia vai da cidade 0 para a N-1.

Saída: Para cada teste deve ser impresso o pedágio mínimo para um percurso com um número par de pedágios. Se não for possível, imprimir -1.

Exemplo de entrada:

```
4 4
0 1 2
1 2 1
1 3 10
2 3 6
```

Exemplo de saída:

```
12
```

2666 - Imposto Real

Contexto: Um reino com cidades $c_1 \ldots c_n$, sendo c_1 a capital, tem um conjunto de estradas estruturados em forma de árvore. O rei mandou recolher os impostos devidos $d_1..d_n$, usando uma carruagem de capacidade r. São dadas as distâncias entre cidades interligadas. Cada cidade tem um cofre muito grande. Qual a distância mínima que a carruagem deve percorrer para recolher os impostos?

Entrada: Um único caso de teste descrito em várias linhas. Na primeira vem os inteiros n, r ($2 \le n \le 10^4$, $1 \le r \le 100$). Na próxima linha vêm n inteiros, os impostos d_i devidos ($0 \le d_i \le 100$).Em seguida n-1 descrições das interligações: 3 inteiros A B C, A e B cidades e C a distância entre elas ($2 < A$, $B \le n$, $1 \le C$, ≤ 100).

Saída: Um inteiro indicando a distância mínima a ser percorrida.

Exemplo de entrada:

```
7 4
0 4 10 9 1 5 0
1 2 1
1 3 2
2 4 3
2 5 1
5 6 2
5 7 3
```

Exemplo de saída:

```
52
```

2962 - Arte Valiosa

Contexto: É dada uma sala de museu de dimensões $M \times N$, onde existe uma porta em $(0, 0)$ e um quadro valioso em (M, N). Foram instalados K detectores de movimentos em posições (x_i, y_i) dadas, cada um tendo um raio de ação igual a s_i. Um ladrão quer roubar o quadro valioso. Conseguirá fazer isso sem ser detectado?.

Entrada: Um único caso de teste descrito em $K+1$ linhas. Na primeira vêm os inteiros M, N, K ($10 \leq M, N \leq 10^4$, $1 \leq K \leq 10^3$). Em seguida K linhas com 3 inteiros, descrevendo sua posição x_i, y_i e seu raio de ação s_i ($0 < x_i < M$, $0 < y_i < N$, $0 < s_i \leq 10^4$).

Saída: Imprimir 'S' se for possível o roubo sem detecção ou 'N', caso contrário.

Exemplo de entrada:

```
10 22 2
4 6 5
6  16  5
```

Exemplo de saída:

```
S
```

1442 - Desvio de Rua

Contexto: É dado um digrafo representando o trânsito de uma cidade. Um trecho de rua vai ser bloqueado. Quer-se saber como contornar o efeito do bloqueio, apenas invertendo o fluxo de algumas ruas ou tornando ruas de mão única em ruas de mão dupla, de forma a que se haja caminho entre quaisquer cruzamentos.

Entrada: Vários casos de teste, terminados por fim de arquivo. Cada teste vem em várias linhas. Na primeira, são informados N, M ($1 \leq N \leq 10^3$, $1 \leq M \leq 10^5$), o número de cruzamentos e trechos de rua, respect. A seguir vêm M linhas indicando os trechos de rua. Cada trecho é informado com 3 inteiros A, B ($1 \leq A, B \leq N$) indicando a ligação e T (1 ou 2), indicando o tipo de trânsito: 1 = mão única, 2=mão dupla. O primeiro trecho é o que vai ser bloqueado.

Saída: Para cada teste indicar o que fazer:

'-' nada precisa ser feito

'*' impossível

'1' inverter o sentido do trânsito de algumas ruas de mão única

'2' tornar alguns trechos de mão única em mão dupla.

Exemplo de entrada:

```
5 7
2 3 1
1 3 2
1 2 1
3 4 1
4 5 1
5 2 1
5 3 1
```

Exemplo de saída:

```
1
```

1391 - Quase o Menor Caminho

Contexto: É dado um digrafo contendo a descrição do mapa de trânsito de uma região: as rotas de trânsito, todas de mão única e com seus tamanhos. Como muitos motoristas usam o GPS para utilizar o caminho mínimo, um motorista quer procurar um caminho alternativo bom para a hora de "rush" entre os pontos s e t que não passe por nenhuma via que possa estar em caminhos mínimos entre esses pontos.

Entrada: Vários casos de teste. Para cada caso de teste é informado n, m, s, t e as m interligações, em termos de 3 inteiros (origem, destino, d =distância). ($2 \leq n \leq 500$, $1 \leq m \leq 10000$, $0 \leq d \leq 1000$).

Saída: Para cada teste deve ser impresso a distância do caminho alternativo de distância mínima. Se não for possível imprimir -1.

Exemplo de entrada:

```
7 9 0 6
0 1 1    0 2 1    0 3 2
0 4 3    1 5 2    2 6 4
3 6 2    4 6 4    3 6 1
```

Exemplo de saída:

```
5
```

2882 - Gasolina

Contexto: No fim de uma greve r refinarias devem abastecer rapidamente p postos. São dados os estoques das refinarias, as demandas dos postos, quais refinarias podem atender quais postos e o tempo de atendimento de cada refinaria ao posto. Quer-se saber qual o tempo mínimo para todos os postos estarem abastecidos.

Entrada: Cada tese inicia c/ 3 inteiros numa linha: p, r ($1 \leq p, r \leq 1000$), número de postos e refinarias e np ($1 \leq np \leq 20000$), o número de pares refinaria-posto. Na próxima linha p inteiros, as demandas dos postos; na terceira linha r inteiros, os estoques das refinarias. Nas próximas np linhas, 3 inteiros I, J, T ($1 \leq T \leq 10^6$) número do posto, número da refinaria e tempo de atendimento.

Saída: Para cada teste deve ser impressa o tempo mínimo de atendimento a todos os postos; -1 se não for possível.

Exemplo de entrada:

```
3  2  5
20 10 10
30 20
1  1  2
2  1  1
2  2  3
3  1  4
3  2  5
```

Exemplo de saída:

```
4
```

1476 - Caminhão

Contexto: Uma cidade é feita de ilhas ligadas por pontes, cada uma com limite máximo de peso dado. Uma empresa tem várias sedes em ilhas dadas e fábricas em ilhas também dadas. São dados vários pares (sede, fábrica) e quer-se saber para cada um desses pares qual o máximo peso que um caminhão pode levar da fábrica para a sede.

Entrada: Vários casos de teste, terminados por fim de arquivo. Cada teste vem em várias linhas. Na primeira, são informados N,($1 \le N \le 2.10^4$), M, ($1 \le M \le 10^5$) e S ($1 \le S \le 5.10^4$), o número de ilhas, pontes e consultas, respect. A seguir vêm M linhas indicando as pontes. Cada ponte é informada com 3 inteiros A, B ($1 \le A, B \le N$) indicando a ligação e C ($\le 10^5$), o limite de peso. A seguir vêm S consultas, sendo cada consulta um par de ilhas.

Saída: Para cada consulta indicar o peso máximo que pode ser transportado por um caminhão entre a fábrica e a sede.

Exemplo de entrada:

```
4 5 3
1 2 30
1 4 40
2 3 20
2 4 50
3 4 10
1 3
1 4
1 2
```

Exemplo de saída:

```
20
40
40
```

1490 - Torres que Atacam

Contexto: O problema das Torres Pacíficas consiste em colocar n torres em um tabuleiro n x n, de tal forma que não se ataquem. Nesta variante, existem peões no tabuleiro, de tal forma que eles podem bloquear ataques. Dado um tabuleiro n x n, com alguns peões posicionados, qual o máximo de torres que não se atacam podem ser colocadas?

Entrada: Cada tese começa com o valor n ($1 \leq n \leq 100$). Em seguida vêm n linhas, descrevendo um tabuleiro, onde 'X' indica um peão posicionado e '.' uma posição livre. Os testes terminam por fim de arquivo.

Saída: Para cada teste deve ser impressa a quantidade de torres que podem ser colocadas no tabuleiro, de forma que não se ataquem.

Exemplo de entrada:

```
5
X....
X....
..X..
.X...
....X
```

Exemplo de saída:

```
5
```