INSTITUTO SUPERIOR TÉCNICO

Análise e Síntese de Algoritmos

Ano Lectivo 2017/2018 1º Teste - versão A

RESOLUÇÃO

I. (2,5 + 2,5 + 2,5 + 2,5 + 2,5 + 2,5 = 15,0 val.)

I.a)

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
$rank[x_i]$	1	0	1	0	2	0	3	0	1	0
$p[x_i]$	x_7	x_7	x_7	x_3	x_7	x_5	x_7	x_7	x_5	x_5

I.b)

	A	B	C	D	E	F	G	H
d/f	1 / 8	15 / 16	10 / 13	3 / 6	2 / 7	4 / 5	9 / 14	11 / 12

$SSCs$:	$\{A,D,E,F\}$	$\{C,G,H\}$	$\{B\}$		

I.c)

Ordem vértices	A	D	E	G	B	H	I	F	C
Custo MST	$5X+3Y$								

I.d)

	A	B	C	D	E	F	G	H
$h()$	-2	0	-2	-4	-7	-1	0	-10

$\widehat{w}(A,B)$	$\widehat{w}(B,E)$	$\widehat{w}(C,E)$	$\widehat{w}(E,G)$	$\widehat{w}(G,H)$
1	3	7	3	12

I.e)

Expressão	$T(n) = 3*T(n/3)+O(\sqrt{n})$
Majorante	$O(n)$

I.f)

	A	B	C	D
$h()$	7	1	7	8

Corte :	$\{s,A,C,D\}/\{B,t\}$	$f(S,T)=$	17

II. (2,0 + 3,0 = 5,0 val.)

II.a) $\langle$ XXX $\rangle$ **II.b)** $\langle$ XXX $\rangle$

I. (2,5 + 2,5 + 2,5 + 2,5 + 2,5 + 2,5 = 15,0 val.)

I.a) Considere o seguinte conjunto de operações sobre conjuntos disjuntos:

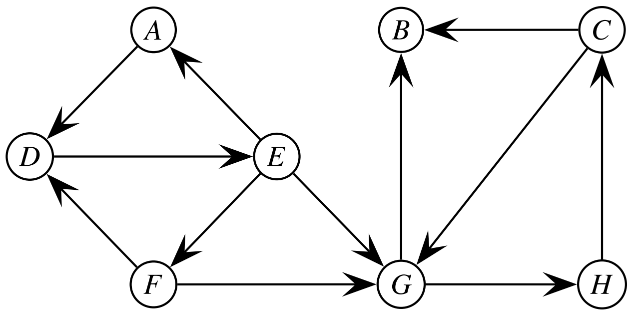
---

for i = *1 to 10* **do**
    Make-Set (x_i)
for i = *1 to 5* **do**
    Union (x_{2i}, x_{2i-1})
Union (x_2, x_7)
Union (x_9, x_5)
Union (x_4, x_8)
Union (x_{10}, x_2)

Use a estrutura em árvore para representação de conjuntos disjuntos com a aplicação das heuristicas de união por categoria e compressão de caminhos. Para cada elemento x_i, indique os valores de categoria ($rank[x_i]$) e o valor do seu pai na árvore ($p[x_i]$).

Nota: Na operação $Make-Set(x)$, o valor da categoria de x é inicializado a 0. Na operação de $Union(x,y)$, em caso de empate, considere que o representante de y é que fica na raíz.

I.b) Considere o grafo dirigido:



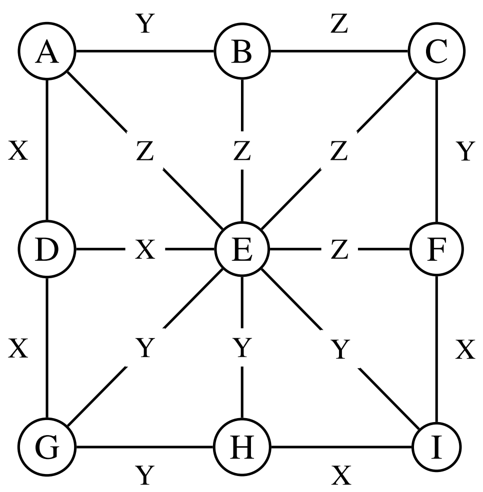
Aplique o algoritmo que utiliza duas travessias em profundidade primeiro (DFS) para encontrar os componentes fortemente ligados do grafo. Considere que na primeira DFS os vértices são considerados por ordem lexicográfica (ou seja, A, B, C...). Em ambas as DFS, os adjacentes são sempre considerados também por ordem lexicográfica.

Indique os tempos de descoberta d e de fim f de cada vértice na **segunda DFS do grafo**.

Indique os componentes fortemente ligados **pela ordem que são descobertos pelo algoritmo**.

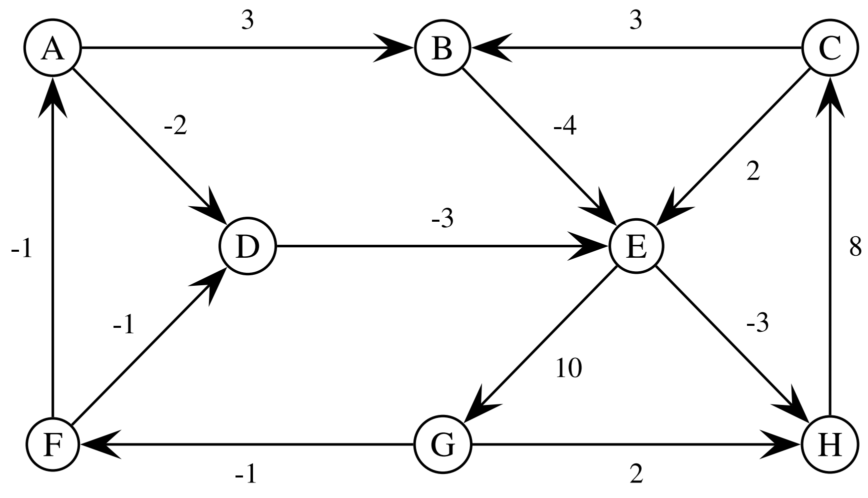
Nota: Neste algoritmo os valores de d começam em 1.

I.c) Considere o grafo não dirigido e pesado da figura onde os pesos dos arcos são inteiros positivos X, Y e Z tal que $X < Y < Z$.



Aplique o algoritmo de Prim ao grafo, considerando o vértice A como origem. Indique a ordem pela qual os vértices são removidos da fila de prioridade durante a execução do algoritmo. Em caso de empate, considere os vértices por ordem lexicográfica.
Indique a expressão que corresponde ao custo de uma MST neste grafo em função de X, Y e Z.

I.d) Considere o grafo dirigido e pesado da figura.



Considere o algoritmo de Johnson. Calcule os valores de $h(u)$ para todos os vértices $u \in V$ do grafo.

Calcule os pesos dos seguintes arcos após a repesagem: (A,B), (B,E), (C,E), (E,G), (G,H).

I.e) Considere a função recursiva:

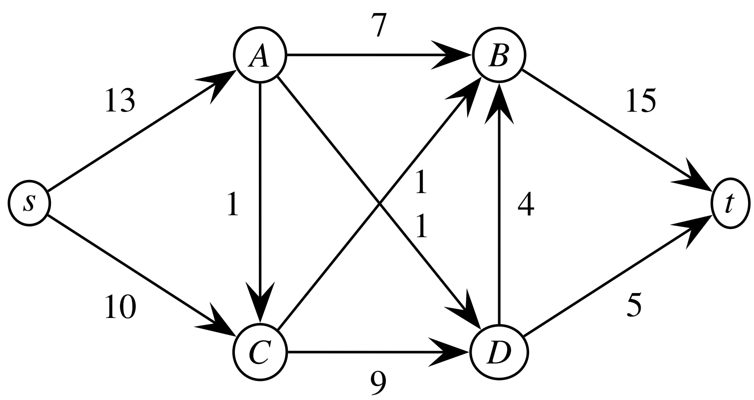
```
int f(int n)
{
  int i = 0, j = 0;
  while(n > i*i) {
    i++;
    j = j + 2;
  }

  if(n > 1)
    i = f(n/3) + f(n/3) + n * f(n/3);

  while (j > 0) {
    i = i + 2;
    j--;
  }
  return i;
}
```

Indique a expressão (recursiva) que descreve o tempo de execução da função em termos do número n, e de seguida, utilizando os métodos que conhece, determine o menor majorante assimptótico.

I.f) Considere a rede de fluxo da figura onde s e t são respectivamente os vértices fonte e destino na rede.

Aplique o algoritmo Relabel-To-Front na rede de fluxo. Considere que a lista de vértices é inicializada por ordem alfabética e que os vizinhos de cada vértice também estão ordenados alfabeticamente. Assim, as listas de vizinhos dos vértices intermédios são as seguintes:

$N[A] = <B, C, D, s>$ $N[B] = <A, C, D, t>$ $N[C] = <A, B, D, s>$ $N[D] = <A, B, C, t>$



Indique a altura final de cada vértice. Indique ainda o corte mínimo da rede e o valor do fluxo máximo.

II. (2,0 + 3,0 = 5,0 val.)

II.a) Considere um grafo $G = (V,E)$ não dirigido e pesado e que $T \subseteq E$ define uma árvore abrangente de menor custo (MST) de G.

Supondo que é removido um arco $(u,v) \in T$ do grafo, proponha um algoritmo que actualiza o conjunto de arcos T tal que T é novamente uma árvore abrangente de menor custo.

Indique a complexidade do algoritmo proposto.

Solução:

Seja $T' = T \setminus \{(u,v)\}$. Considerando um qualquer vértice s do grafo como origem, aplica-se uma pesquisa em profundidade primeiro (DFS) para identificar os vértices atingíveis a partir de s usando apenas os arcos de T'.

Seja S o conjunto dos vértices atingíveis a partir de s e $R = V \setminus S$. Seja $(x,y) \in E$ o arco de menor peso tal que $x \in S$ e $y \in R$. Nesse caso, podemos formar uma nova árvore abrangente de menor custo adicionando (x,y) a T'. A complexidade do algoritmo proposto é $O(E)$.

II.b) Suponha a situação em que há duas concentrações de manifestantes: uma a favor da saída do Reino Unido da União Europeia e outra contra. Após a concentração, os manifestantes irão percorrer algumas ruas previamente definidas de forma a que os percursos não se cruzam.

No entanto, o Sr. Caracol, comandante da polícia, quer certificar-se que não existem desvios nos percursos já definidos tal que os manifestantes das duas concentrações se encontrem numa mesma rua.

O Sr. Caracol tem um mapa da cidade sob a forma de um grafo $G = (V,E)$ não dirigido e pesado. Cada arco (u,v) do grafo representa uma rua e o peso $w(u,v)$ o número de polícias necessários para impedir os manifestantes de usarem a rua (u,v).

Considere que $p_1 =< u_1, u_2 \ldots u_k >$ define o percurso de uma das manifestações e que $p_2 =< v_1, v_2 \ldots v_m >$ define o percurso da outra manifestação. O Sr. Caracol pretende colocar polícias a bloquear algumas ruas de forma a que se os manifestantes se desviarem dos percursos definidos, então não se irão encontrar com os manifestantes opostos.

Defina um algoritmo que ajude o Sr. Caracol a definir quais as ruas a bloquear tal que o número de polícias necessário para esta operação seja mínimo. Indique a complexidade da solução proposta.

Solução:

O problema apresentado corresponde a encontrar um corte mínimo num grafo não dirigido que pode ser resolvido usando uma rede de fluxo $G' = (V',E')$ da seguinte forma:

- Definem-se os vértices da rede de fluxo como sendo $V' = V \cup \{s,t\}$;
- Para cada arco $(u,v) \in E$ adicionam-se os arcos (u,v) e (v,u) a E' com capacidade $w(u,v)$;
- Para cada vértice $u_i \in p_1$ adiciona-se um arco (s,u_i) a E' com capacidade $+\infty$;
- Para cada vértice $v_j \in p_2$ adiciona-se um arco (v_j,t) a E' com capacidade $+\infty$;

Os arcos que definem o corte mínimo da rede de fluxo correspondem às ruas a bloquear, minimizando o número de polícias necessários para separar os conjuntos de manifestantes. A complexidade do algoritmo proposto é $O(V^3)$ se aplicarmos o algoritmo Relabel-To-Front.