TERCEIRA PROVA DE ALGORITMOS EM GRAFOS
BCC, 1o. SEMESTRE DE 2005

1. [3 pontos] Seja $G = (V, E)$ um grafo conexo e $c\colon E \to \mathbb{R}$ uma função custo definida nas arestas de G.
   (i) Seja $A \subset E$ um conjunto de arestas contida em alguma árvore geradora mínima (AGM) de (G, c). Seja $S \subset V$ tal que o corte $E(S, V \setminus S)$ é disjunto de A. Seja $e \in E(S, V \setminus S)$ tal que $c(e) = \min\{c(f)\colon f \in E(S, V \setminus S)\}$, isto é, a aresta e tem custo mínimo dentre as arestas do corte $E(S, V \setminus S)$. Prove que $A \cup \{e\}$ está contida em uma AGM de (G, c).
   (ii) Considere o algoritmo abaixo:
   (1) $A \leftarrow \emptyset$
   (2) Enquanto $|A| < |V| - 1$, faça:
   (2.1) $C \leftarrow$ uma componente conexa arbitrária de $G[A] = (V, A)$
   (2.2) $e \leftarrow$ uma aresta de custo mínimo que liga C a alguma outra componente de $G[A]$
   (2.3) $A \leftarrow A \cup \{e\}$
   (3) Devolva A
   Prove que, ao término da execução do algoritmo acima, $G[A]$ é uma AGM de (G, c).
2. [3 pontos] Seja $G = (V, E)$ um grafo conexo e $c\colon E \to \mathbb{R}$ uma função custo definida sobre as arestas de G. Dado um subgrafo H de G, pomos $c_{\max}(H) = \max\{c(e)\colon e \in E(H)\}$, isto é, $c_{\max}(H)$ é o custo máximo das arestas em H.
   (i) Prove que se T é uma AGM de (G, c) e T' é uma árvore geradora arbitrária de G, então $c_{\max}(T) \le c_{\max}(T')$. [*Sugestão.* Para $e \in E(T)$, considere as duas subárvores $T_1^{(e)}$ e $T_2^{(e)}$ obtidas pela remoção de e de T.]
   (ii) Prove que se T e T' são duas AGMs de (G, c), então $c_{\max}(T) = c_{\max}(T')$.
   (iii) Descreva um algoritmo eficiente para encontrar uma árvore geradora T^* de G com $c_{\max}(T^*)$ mínimo. Diga qual é a complexidade de seu algoritmo. Você acha que existem algoritmos mais eficientes que o seu para resolver este problema? Justifique suas respostas.
3. [2.5 pontos] Considere a implementação do algoritmo de Dijkstra visto em sala (Programa 21.1):

```
#define GRAPHpfs GRAPHspt
#define P (wt[v] + t->wt)
void GRAPHpfs(Graph G, int s, int st[], double wt[])
  { int v, w;  link t;
    PQinit(); priority = wt;
    for (v = 0; v < G->V; v++)
      { st[v] = -1; wt[v] = maxWT; PQinsert(v); }
    wt[s] = 0.0; PQdec(s);
    while (!PQempty())
      if (wt[v = PQdelmin()] != maxWT)
        for (t = G->adj[v]; t != NULL; t = t->next)
          if (P < wt[w = t->v])
            { wt[w] = P; PQdec(w); st[w] = v; }
  }
```

Seja G o grafo dirigido com 6 vértices e 8 arcos com custos nos arcos dado por

```
0: 1(2) 2(1)
1: 2(3) 3(3)
2: 4(1)
3: 5(2)
4: 3(2) 5(5)
5:
```

Na notação acima, por exemplo, "`0: 1(2)`" significa que há um arco de `0` para `1` de custo `2`. No que segue, use ∞ para representar `maxWT`.

(i) Desenhe o grafo dirigido G.

(ii) Execute o Programa 21.1 no grafo G dado acima, com $s = 0$, ilustrando claramente a evolução do algoritmo através de diagramas convenientes (uns 6 diagramas devem bastar). Em seus diagramas, coloque os valores de `wt[]` de forma conveniente.

(iii) Escreva explicitamente os valores finais dos vetores `st[]` e `wt[]`. Mostre na sua figura final de (ii) a estrutura que o vetor `st[]` representa.

(iv) Seja G^* o grafo não-dirigido (com custos nas arestas) obtido a partir de G ignorando-se as orientações dos arcos de G. Assim, G^* tem 6 vértices e 8 arestas. Encontre uma AGM de G^*.

(v) Prove ou desprove: uma árvore de caminhos mínimos de um grafo dirigido G fortemente conexo com custos nos arcos (como, por exemplo, obtida pelo algoritmo de Dijkstra) é sempre uma AGM do grafo não-dirigido G^* obtido como em (iv) acima. Justifique sua resposta claramente.

4. [2.5 pontos] Seja G o grafo dirigido com 6 vértices e 10 arcos, com *capacidades* nos arcos, dado abaixo:

```
0: 1(16) 2(13)
1: 2(10) 3(12)
2: 1(4) 4(14)
3: 2(9) 5(20)
4: 3(7) 5(4)
5:
```

Na notação acima, por exemplo, "`0: 1(16)`" significa que há um arco de `0` para `1` de capacidade `16`. No que segue, consideramos fluxos de $s = 0$ para $t = 5$ no grafo G acima.

(i) Desenhe o grafo dirigido G.

(ii) Considere o fluxo de s para t em G dado por:

```
0: 1(11) 2(8)
1: 2(0) 3(12)
2: 1(1) 4(11)
3: 2(4) 5(15)
4: 3(7) 5(4)
5:
```

Acima, por exemplo, "`0: 1(11)`" significa que o fluxo no arco de `0` para `1` é `11`. Encontre um caminho aumentador para o fluxo acima.

(iii) Encontre um fluxo de valor maior que o fluxo dado em (ii), usando o caminho aumentador que você encontrou em (ii) para aumentar aquele fluxo.

(iv) Encontre um fluxo máximo de s para t no G dado acima. Encontre um corte mínimo que separa s de t em G. Dê um argumento para provar que seu fluxo é de fato máximo e seu corte é de fato mínimo.

(v) Um arco é *vital* se a redução de sua capacidade (por menor que seja essa redução) reduz o valor do fluxo máximo. Determine todos os arcos vitais em G dado acima. Justifique sua resposta.