Algoritmos de Aproximação
SEGUNDO SEMESTRE DE 2012
Terceira Prova – 26 de novembro

Nome do aluno: ______________________________________ **Curso**: ____________

Assinatura: ____

No. USP: ______________________ **Professor**: ______________________________

Instruções

1. Não destaque as folhas deste caderno.
2. A prova pode ser feita a lápis.
3. A legibilidade também faz parte da nota!
4. A prova consta de 4 questões, 3 disponibilizadas agora, para você fazer aqui. Verifique antes de começar a prova se o seu caderno de questões está completo.
5. Não é permitido o uso de folhas avulsas para rascunho.
6. Não é necessário apagar rascunhos no caderno de questão mas especifique qual é a resposta e qual é o rascunho.
7. A prova é sem consulta.
8. Você pode usar, sem ter que escrever, algoritmos vistos em aula. Neste caso, antes de usá-lo, deixe claro qual é o protótipo do algoritmo, o que ele devolve/faz, e quanto tempo ele consome em função de sua entrada.

Não escrever nesta parte da folha

Questão	Nota	Observação
1		
2		
3		
4		
Total		

Boa prova!

1. [3,5 pontos]
   No *Problema do Rotulamento Uniforme*, são dados um grafo $G = (V, E)$, custos $c_e \geq 0$ para cada aresta e em E, e um conjunto de rótulos L que podem ser atribuídos aos vértices de V. Existe um custo não-negativo $c_v^i \geq 0$ para atribuir o rótulo i em L ao vértice v, e uma aresta $e = uv$ paga o custo c_e se u e v receberem rótulos distintos. O objetivo do problema é atribuir um rótulo a cada vértice de V de modo a minimizar o custo total.

   Segue uma formulação linear inteira do problema. Seja x_v^i uma variável binária que vale 1 se o vértice v recebe o rótulo i em L, e 0 caso contrário. Seja z_e^i uma variável binária que vale 1 se exatamente um de seus extremos recebe rótulo i, e 0 caso contrário. Então a formulação linear inteira é, encontrar x e z que

$$
\begin{array}{lcl}
\text{minimizem} & \frac{1}{2} \sum_{e \in E} c_e \sum_{i \in L} z_e^i + \sum_{v \in V, i \in L} c_v^i x_v^i & \\
\text{sujeitos a} & \sum_{i \in L} x_v^i = 1 & \text{para todo } v \text{ em } V, \\
& z_e^i \geq x_u^i - x_v^i & \text{para toda } e = uv \text{ em } E, \text{ e } i \text{ em } L, \\
& z_e^i \geq x_v^i - x_u^i & \text{para toda } e = uv \text{ em } E, \text{ e } i \text{ em } L, \\
& z_e^i \in \{0, 1\} & \text{para toda } e \text{ em } E \text{ e } i \text{ em } L, \\
& x_v^i \in \{0, 1\} & \text{para todo } v \text{ em } V \text{ e } i \text{ em } L.
\end{array}
$$

   (a) Prove que a formulação linear inteira acima modela o problema.

   Considere agora o seguinte algoritmo. Primeiro, o algoritmo resolve a relaxação linear do programa inteiro acima. O algoritmo então procede em fases. Em cada fase, ele escolhe um rótulo i em L aleatoriamente com distribuição uniforme, e escolhe independentemente um número aleatório α em $[0, 1]$, com distribuição uniforme. Para cada vértice v em V que ainda não tem rótulo, o algoritmo atribui rótulo i a v caso $\alpha \leq x_v^i$.

   (b) Suponha que o vértice v não teve um rótulo atribuído a ele ainda. Prove que a probabilidade dele ter o rótulo i em L atribuído a ele na próxima fase é exatamente $x_v^i/|L|$, e que a probabilidade que ele receba um rótulo na próxima fase é $1/|L|$. Ademais, prove que a probabilidade que o algoritmo atribua um rótulo i a v é exatamente x_v^i.

   (c) Dizemos que uma aresta e é *separada* por uma fase se os dois extremos de e não tinham rótulo no início da fase, e exatamente um deles é rotulado nesta fase. Suponha que os dois extremos de e não estejam rotulados ainda. Prove que a probabilidade de uma aresta e ser separada na próxima fase é exatamente $\frac{1}{|L|} \sum_{i \in L} z_e^i$.

   (d) Prove que a probabilidade dos extremos de uma aresta e receberem rótulos distintos é no máximo $\sum_{i \in L} z_e^i$.

   (e) Prove que o algoritmo é uma 2-aproximação para o Problema do Rotulamento Uniforme.

2. [3,0 pontos]
Lembre-se do problema do corte direcionado máximo (MAX DICUT): dado um digrafo $D = (V, A)$ e um peso não-negativo w_{ij} para cada arco ij em A, encontrar uma partição de V em conjuntos U e $W = V \setminus U$ que maximize o peso total dos arcos de U para W, ou seja, arcos (i, j) com i em U e j em W.

   (a) Expresse o MAX DICUT como um programa quadrático cuja única restrição é $y_i \in \{-1, 1\}$ para todo i e com uma função objetivo quadrática nos y_i. *Dica:* Use uma variável y_0 que indica se o valor -1 ou 1 para y_i significa que o vértice i está ou não no conjunto U.

   (b) Escreva uma α-aproximação para o MAX DICUT usando a relaxação vetorial do programa quadrático do item (a). Encontre o melhor valor de α que você puder. *Dica:* Você pode usar o lema visto em aula que diz que: para todo x em $[-1, 1]$, $\frac{1}{\pi} \arccos(x) \geq 0.878 \, \frac{(1-x)}{2}$.

3. [1,5 pontos]
Esta questão está dividida em duas partes. A segunda parte (questão 4) vale 2,0 pontos e você receberá no final da prova. Você deve entregar a resolução da segunda parte desta questão na aula de quarta-feira.

Considere o *problema do multicorte em árvores*: dada uma árvore $T = (V, E)$, k pares de vértices $\{s_i, t_i\}$, e custos $c_e \geq 0$ para cada aresta e em E, encontrar um conjunto de arestas F de T de custo mínimo cuja remoção desconecta os k pares de vértices dados. Ou seja, s_i e t_i ficam em componentes distintas do grafo $T' = (V, E \setminus F)$, para todo i.

Seja P_i o conjunto das arestas no único caminho entre s_i e t_i em T. Então podemos formular o problema como o seguinte programa linear inteiro: encontrar um vetor x que

$$\begin{array}{lll} \text{minimize} & \sum_{e \in E} c_e \, x_e & \\ \text{sujeito a} & \sum_{e \in P_i} x_e \geq 1 & \text{para } i = 1, \dots, k, \\ & x_e \in \{0, 1\} & \text{para todo } e \text{ em } E. \end{array}$$

Suponha que a árvore T tem uma raiz r, e seja $\mathrm{prof}(v)$ a *profundidade* de v em relação a r, ou seja, o número de arestas no caminho de r até v em T. Seja $\mathrm{lca}(s_i, t_i)$ o vértice v de profundidade mínima no caminho entre s_i e t_i em T (lca - *lowest common ancestor*).

(a) Escreva o dual da relaxação linear do programa acima e um primeiro algoritmo primal-dual para o problema do multicorte em árvores que escolha, a cada iteração, uma variável dual associada a uma restrição violada do primal para um i tal que $\mathrm{prof}(\mathrm{lca}(s_i, t_i))$ é máximo.

(b) Mostre as desigualdades que você gostaria de provar para deduzir que seu algoritmo é uma 2-aproximação, explicando as que você sabe que valem.

(c) Mostre que, sem o passo de limpeza ao final do algoritmo (que joga fora algumas das arestas escolhidas durante a primeira fase do algoritmo), o seu algoritmo não é uma 2-aproximação para o problema. Se possível, mostre que seu algoritmo não é uma α-aproximação para nenhum $\alpha > 1$ constante.

4. [2,0 pontos]
Considere o *problema do multicorte em árvores*: dada uma árvore $T = (V, E)$, k pares de vértices $\{s_i, t_i\}$, e custos $c_e \geq 0$ para cada aresta e em E, encontrar um conjunto de arestas F de custo mínimo cuja remoção desconecta os k pares de vértices dados. Ou seja, s_i e t_i ficam em componentes distintas do grafo $T' = (V, E \setminus F)$, para todo i.

Seja P_i o conjunto das arestas no único caminho entre s_i e t_i em T. Um conjunto F de arestas de T é *viável* se s_i e t_i ficam em componentes distintas do grafo $T' = (V, E \setminus F)$ para todo i.

Suponha que a árvore T tem uma raiz r, e seja $\text{prof}(v)$ a *profundidade* de v em relação a r, ou seja, o número de arestas no caminho de r até v em T. Seja $\text{lca}(s_i, t_i)$ o vértice v cuja profundidade mínima no caminho entre s_i e t_i em T.

Considere a relaxação da formulação linear inteira da questão anterior e seu dual:

$$\begin{array}{rll} \text{minimize} & \sum_{e \in E} c_e \, x_e & \\ \text{sujeito a} & \sum_{e \in P_i} x_e \geq 1 & \text{para } i = 1, \ldots, k, \\ & x_e \geq 0 & \text{para todo } e \text{ em } E. \end{array}$$

$$\begin{array}{rll} \text{maximize} & \sum_{i=1}^{k} y_i & \\ \text{sujeito a} & \sum_{i : e \in P_i} y_i \leq c_e & \text{para cada } e \text{ em } E, \\ & y_i \geq 0 & \text{para } i = 1, \ldots, k. \end{array}$$

Considere o seguinte algoritmo primal-dual para o problema do multicorte em árvores.

Primal-Dual (G, s, t, k, c)
$F \leftarrow \emptyset \qquad y \leftarrow 0$
enquanto F não é viável **faça**
  seja i tq s_i e t_i estão na mesma componente de $T' = (V, E \setminus F)$ e $\text{lca}(s_i, t_i)$ é máximo
  $\epsilon \leftarrow \min\{c_e - \sum_{j : e \in P_j} y_j \ : \ e \in P_i\}$
  $f \leftarrow \arg\min\{c_e - \sum_{j : e \in P_j} y_j \ : \ e \in P_i\}$
  $y_i \leftarrow y_i + \epsilon$
  $F \leftarrow F \cup \{f\}$
$F' \leftarrow F$
para cada aresta e em F em ordem reversa à inclusão em F **faça**
  se $F' \setminus \{e\}$ é viável
    então $F' \leftarrow F' \setminus \{e\}$
devolva F'

(a) Prove que, para cada i tal que $y_i > 0$, existem no máximo duas arestas de F' em P_i. (*Dica:* Seja $u_i = \text{lca}(s_i, t_i)$. Prove que, se $y_i > 0$, então existe no máximo uma aresta de F' no caminho de s_i a u_i, e no máximo uma aresta de F' no caminho de u_i a t_i.)

(b) Deduza disso que o algoritmo acima é uma 2-aproximação para o problema do multicorte em árvores.

(c) Mostre que, se a remoção das aresta de F' for feita em uma ordem arbitrária, (a) pode deixar de valer.