MC658 - Análise de Algoritmos III
Lista de Exercícios de Algoritmos Aproximados

1. Apple e Haken mostraram em 1976 que todo grafo planar pode ser colorido com 4 cores. Projete uma 1-aproximação absoluta para o problema de coloração de vértices em grafos planares.

2. Mostre que não existe k-aproximação absoluta para o TSP com pesos inteiros nas arestas a menos que P = NP.

3. Mostre que não pode existir k-aproximação absoluta (k constante) para o problema clique máxima assumindo que $P \neq NP$. **Dica:** Dado instância $G = (V, E)$ use grafo G^m, composto por m cópias G_i de G, e existe arestas entre vértices de cópias distintas (para $v \in G_i$ e $u \in G_j$ com $i \neq j$ vale que $(u, v) \in E^m$).

4. Sobre o Vertex Cover

   (a) Mostre que o algoritmo 2-aproximado visto em aula para o Vertex Cover é justo. Ou seja, mostre uma instância de entrada para o algoritmo tal que a resposta deste é 2 vezes pior do que uma solução ótima.

   (b) Considerando o algoritmo visto em aula mostre que ele sempre retorna um emparelhamento maximal.

   (c) Projete um algoritmo eficiente ótimo para o problema Vertex-Cover considerando que os dados de entrada são sempre uma árvore.

   (d) Considere a seguinte heurística para o problema: A cada iteração pegue o vértice com maior grau no grafo como parte da cobertura. Retire as arestas incidentes a este e repita o processo com o grafo restante. Mostre que esta heurística não pode ter aproximação menor do que $(2 - \epsilon)$ dando uma instância de entrada tal que a solução retornada é aproximadamente 2 vezes pior que a solução ótima.
   Dica: Considere um grafo bipartido onde em uma das partições os vértices tem todos os mesmos graus.

   (e) Há uma relação de redução do problema da Clique máxima para o Vertex-Cover. Existe clique de tamanho $|C|$ em G se e somente se existe uma cobertura por vértices de tamanho $|V| - |C|$ em $\overline{G}$ (grafo complementar de G). Isto implica que se tivermos uma aproximação constante (Exemplo 2) para o Vertex-Cover também teremos uma aproximação para o problema Clique máxima? Justifique.

5. Mostre que o algoritmo de Graham para o problema de escalonamento de tarefas em máquinas paralelas (visto em aula) retorna uma solução com valor **estritamente** menor do que 2OPT.

6. Sobre o algoritmo para o TSP-métrico.

   (a) Seja G um grafo completo com pelo menos 3 vértices e que satisfaça a desigualdade triangular. Mostre que para quaisquer par de vértices (u, v) de G temos que $c(u, v) \geq 0$.

   (b) Mostre que o problema de decisão do TSP-métrico é NP-Completo.

   (c) Considere a seguinte heurística para o Problema: Inicialmente temos um ciclo composto de um único vértice. A cada passo identifique o vértice mais próximo do ciclo que ainda não pertença a este. Suponha que o vértice u é o mais próximo e está mais próximo a um vértice v. Inclua u no ciclo logo após v. Mostre que esta heurística é um algoritmo 2-aproximado para o TSP-métrico.

7. No algoritmo visto em aula para o MAX-3CNF, mostre que este permanece uma $7/8$-aproximação probabilística mesmo que em uma cláusula apareça um literal e sua negação.

8. Mostre que o algoritmo para o problema MAX-3CNF visto em aula é uma $1/2$ aproximação probabilística quando aplicado ao problema MAX-CNF (neste problema cada cláusula tem pelo menos um literal mas não há um limite superior para o número de literais que aparecem em uma cláusula).

9. No problema MAX-CUT temos um grafo $G = (V, E)$ de entrada. Dado $S \subseteq V$ definimos o corte S em G como sendo as arestas de G que possuem um vértice em S e outro em $V - S$. O problema é determinar um corte S que maximize o número de arestas. Suponha o seguinte algoritmo: Para cada vértice atribua este de forma independente dos demais a S com probabilidade $1/2$. Mostre que este algoritmo é uma 2-aproximação probabilística.

10. Considere o problema *bin packing* unidimensional. Temos uma lista de itens $\{1, \ldots, n\}$, cada item i com tamanho $0 \leq t(i) \leq 1$. Deve-se empacotar todos os itens em recipientes de tamanho 1.

    a Mostre que o algoritmo Next-Fit (NF) descrito abaixo para o problema *bin packing* é tal que para qualquer instância I vale que $NF(I) \leq 2OPT(I)+1$. Algoritmo NF: Inicialmente cria-se um *bin* B_1 e faz-se $j = 1$: Para cada item e repita: empacota e no bin B_j; caso não seja possível cria-se um novo bin B_{j+1}, faça $j \leftarrow j + 1$, e empacote e neste novo bin.
    Dica: Use como o limitante para o ótimo a soma dos tamanhos dos itens $\sum_{i=1}^{n} t(i)$.

    b Mostre que o algoritmo First-Fit (FF), descrito a seguir é uma 2-aproximação para o problema. Algoritmo FF: Inicialmente cria-se um *bin* B_1 e faz-se $j = 1$. Para cada item e da lista repita: empacote e no primeiro recipiente (na ordem em que foram criados) que caiba o item; caso nenhum *bin* criado possa empacotar e, então crie um novo *bin* B_{j+1}, faça $j \leftarrow j + 1$, e empacote e neste novo bin.

    c Mostre que não existe algoritmo com fator de aproximação menor que $3/2$ para o problema *bin packing* a menos que P=NP.
    Dica: use o problema da partição.

11. (*) Mostre que o algoritmo abaixo é um algoritmo $3/2$-aproximado para o TSP métrico. Considere o seguinte algoritmo para o TSP métrico.

    (a) Determine uma árvore geradora mínima T de G como antes.

    (b) Seja I o conjunto dos vértices de grau ímpar na **árvore** T.

    (c) Encontre um emparelhamento perfeito de peso mínimo em $G[I] := G - (V - I)$, digamos M. Note que $|I|$ é par.

    (d) Seja $H := T \cup M$. Note que todos os vértices de H têm grau par.

    (e) Encontre um ciclo euleriano (ou seja, um ciclo que passa por todas as arestas exatamente uma vez) de H (isso é sempre possível pois todos os vértices de H têm grau par).

    (f) Visitando os vértices na mesma ordem do ciclo euleriano, aplique o *shortcuts* como antes para obter um circuito hamiltoniano C de G. Note que o custo de C é menor ou igual ao custo do ciclo euleriano.

12. (*) O problema da árvore de Steiner consiste no seguinte: dado um grafo G com custos nas arestas satisfazendo a desigualdade triangular e um subconjunto S de vértices de G, encontrar

uma árvore em G que conecta S e que tenha custo mínimo. Note que a árvore pode ter vértices fora de S. Descreva um algoritmo de 2-aproximação para o problema. Onde você usou o fato dos custos satisfazerem a desigualdade triangular?

13. O TSP com pesos 1 e 2 consiste em encontrar um circuito hamiltoniano em um grafo completo G onde todas as arestas possuem custo 1 ou 2. Apresente um algoritmo 4/3-aproximado para o problema TSP com pesos 1 e 2. **Dica:** Use um 2-emparelhamento de custo mínimo. Um 2-emparelhamento de G é um subgrafo gerador de G onde todos os vértices possuem grau 2 (note que ele não precisa ser conexo). Existem algoritmos polinomiais que acham um 2-emparelhamento de custo mínimo em um grafo qualquer (você não precisa saber como fazer isso para resolver a questão).

14. Considere o problema do *hitting set*: dados subconjuntos $S_1, \ldots, S_n$ de um conjunto S, encontrar um subconjunto X de S tal que $|X \cap S_i| \geq 1$ (ou seja, X intercepta todo subconjunto) e que tenha tamanho mínimo. Projete um algoritmo de k_{max}-aproximação para o *hitting set* onde k_{max} é o tamanho do maior subconjunto S_i. **Dica:** Formule este problema com PL.

15. Considere o problema MAX-Sub-Set-Sum: Dados um conjunto de elementos inteiros positivos $S = \{x_1, \ldots, x_n\}$ e um inteiro positivo t deve-se achar subconjunto $S' \subseteq S$ tal que a soma dos elementos em S' seja máxima mas menor ou igual a t. Dado o FPTAS para o problema da mochila visto em aula, projete um FPTAS para este problema.