Universidade Estadual de Campinas – Instituto de Computação

MO417A – Complexidade de Algoritmos – 1º Semestre de 2020

https://www.ic.unicamp.br/~lehilton/mo417a

Reduções entre problemas

Conceitos sobre redução

Questão 1. Sejam P_1 e P_2 dois problemas tais que $P_1 \preccurlyeq_n P_2$ e suponha que P_1 tem cota inferior $\Omega(n \log n)$, onde n é um parâmetro que mede o tamanho da entrada do problema P_1. Quais das seguintes afirmações são verdadeiras? Justifique cuidadosamente as suas respostas.

- ***(a)*** C E $\Omega(n \log n)$ também é cota inferior para P_2.
- ***(b)*** C E Todo algoritmo que resolve P_1 também pode ser usado para resolver P_2.
- ***(c)*** C E Todo algoritmo que resolve P_2 também pode ser usado para resolver P_1.
- ***(d)*** C E O problema P_2 pode ser resolvido no pior caso em tempo $O(n \log n)$.

Questão 2. Sejam P_1 e P_2 dois problemas tais que um deles tenha cota inferior $\Omega(n^k)$, para algum $k > 1$, e o outro é solúvel em tempo $O(n \log n)$. Se P_1 é redutível a P_2 em tempo linear, diga qual é qual. O parâmetro n denota o tamanho da entrada dos dois problemas.

Questão 3. Sejam A e B dois problemas tais que $A \preccurlyeq_{n \log n} B$ e suponha que A tem cota inferior $\Omega(n^{3/2})$, onde n é um parâmetro que mede o tamanho de uma instância I_A de A. Suponha que dada I_A, a redução obtém uma instância I_B de B. Quais das seguintes afirmações são verdadeiras? Justifique.

- ***(a)*** C E A notação $A \preccurlyeq B$ significa que o pior algoritmo para A é mais rápido que o melhor algoritmo para B.
- ***(b)*** C E Se existe algoritmo $O(n^{3/2})$ para B, então o problema A pode ser resolvido em tempo $O(n \log n)$.
- ***(c)*** C E Se a instância I_B tem tamanho $\Theta(n)$, então $\Omega(n^{3/2})$ também é cota inferior para B.
- ***(d)*** C E Se não existe um algoritmo que resolve A, então também não existe um algoritmo que resolve B.
- ***(e)*** C E Se existe um algoritmo que resolve A, então também existe um algoritmo que resolve B.

Reduções entre problemas

Questão 4. Diz-se que um ponto $p = (x_p, y_p)$ do plano *domina* um outro ponto *distinto* $q = (x_q, y_q)$ do plano se $x_p \geq x_q$ e $y_p \geq y_q$. Um ponto p é *maximal* em relação a um conjunto de pontos P se $p \in P$ e nenhum outro ponto de $P - \{p\}$ domina p (note que isto *não* significa que p domina todos os pontos de $P - \{p\}$).

Projete um algoritmo de complexidade $O(n \log n)$ para encontrar todos os pontos maximais de um conjunto P de n pontos no plano.

Exemplo: suponha que $P = \{(0, n-1), (1, n-2), \ldots, (n-2, 1), (n-1, 0)\}$. Quais sãos os pontos maximais de P?

Questão 5. Considere o seguinte problema: dados n intervalos (fechados) na reta real, definidos pelos seus pontos de início e de fim, projete um algoritmo que lista todos os intervalos que estão contidos dentro de pelo menos um dos outros intervalos passados na entrada. O seu algoritmo deve ter complexidade $O(n \log n)$.

Questão 6. Denote por `MAXIMAL` o problema do exercício 4 e por `INTERVAL` o problema do exercício 5. Encontre uma redução de complexidade linear de `MAXIMAL` para `INTERVAL`.

É possível usar o algoritmo desenvolvido no exercício anterior e a redução proposta por você para projetar um algoritmo para `MAXIMAL`? Em caso afirmativo, como se compara a complexidade deste algoritmo com a do algoritmo do exercício 4?

Questão 7. Encontre uma redução de complexidade linear de `INTERVAL` para `MAXIMAL`.

Questão 8. Usando o conceito de dominância entre pontos do exercício 4, pode-se definir os *Pareto*s de um dado conjunto não vazio de pontos $P = \{p_1, \ldots, p_n\}$ no plano da seguinte forma:

(i) o *Pareto* 1 de P, denotado por P_1, é o conjunto de pontos maximais de P;

(ii) para $i \geq 2$, o *Pareto* i de P, denotado por P_i, é o conjunto de pontos maximais de $P \setminus (P_1 \cup \ldots \cup P_{i-1})$.

Chamemos de *índice de Pareto* de P o maior valor de i para o qual o *Pareto* i é não vazio. Denotemos por $i(P)$ este valor.

Dado um conjunto P como acima, considere o problema de encontrar os $i(P)$ primeiros Paretos de P. Projete um algoritmo $O(n \log n)$ para este problema.

Questão 9. Encontre uma redução polinomial do problema de ordenação de um vetor de n elementos para o problema `PARETO` do exercício anterior. A sua redução deve ter complexidade $O(n)$.

Pergunta-se: esta redução prova que o algoritmo do exercício anterior é ótimo (do ponto de vista de complexidade computacional)? Justifique sua resposta.

Questão 10. Uma matriz quadrada é dita ser *triangular inferior* (*superior*) se todos os seus elementos não nulos estiverem na diagonal principal ou abaixo (acima) dela.

Seja `MMIS` o problema de multiplicar uma matriz triangular inferior por uma matriz triangular superior e `MMQ` o problema de multiplicar duas matrizes quadradas arbitrárias.

Seja $T(n)$ a complexidade de um algoritmo ótimo para resolver `MMIS` quando as matrizes passadas na entrada tem ordem n. Suponha que $T(cn) \in O(T(n))$ para toda constante $c > 0$.

Mostre que `MMIS` é pelo menos tão difícil quanto `MMQ` no sentido de que ambos têm a mesma cota inferior (supondo o modelo de computação usual).

Questão 11. Seja $G = (V, E)$ um grafo direcionado com custo $c(v) \geq 0$ para cada vértice de V. O custo de um caminho direcionado entre vértices s e t é a soma dos custos de todos os *vértices internos* desse caminho. Formalmente, o custo de um caminho $(s, x_1, x_2, \ldots, x_k, t)$ é definido como $\sum_{i=1}^{k} c(x_i)$. Por exemplo, se (s, v) é uma aresta do grafo, então o custo desse caminho é *zero*.

Deseja-se encontrar um caminho de custo mínimo de s para todos os vértices de $V \setminus \{s\}$.

Encontre uma redução do problema de construir uma árvore de caminhos mínimos (com custos nos vértices) a partir de s para o problema de construir uma árvore de caminhos mínimo usual (com custos nas arestas).

Questão 12. (difícil) Seja $G = (V, E)$ um grafo não direcionado tal que pra cada vértice v do grafo temos associado uma função $b(v) \leq grau(v)$. Um b-emparelhamento é um subconjunto de E tal que cada vértice v não tem mais do que $b(v)$ arestas incidentes a ele. Em outras palavras, um b-emparelhamento é um subgrafo gerador de G onde cada vértice v tem grau menor ou igual a $b(v)$. Um b-emparelhamento máximo é aquele que tem o maior número de arestas possível. Reduza o problema de se achar um b-emparelhamento máximo ao problema de se achar um emparelhamento máximo em um grafo.

Questão 13. Considere uma rede de computadores, que são ligados aos pares por cabos bidirecionais de diferentes tipos e comprimentos. Dependendo do tipo e do comprimento, uma ligação tem uma taxa de transmissão, dada em MBit/s. A velocidade de conexão entre dois computadores A e B depende do cabo mais lento por que passam os dados no caminho mais rápido de A a B. A velocidade da rede é a menor velocidade de conexão entre todos pares de computadores. Escreva um algoritmo para calcular a velocidade da rede e encontrar um par A, B cuja velocidade de conexão é igual à velocidade da rede.

Questão 14. Seja S um conjunto de n pontos distintos do plano e $G = (S, E)$ o grafo completo onde cada vértice de G corresponde a um ponto de S. Além disso, suponha que a cada aresta (u, v) está associado um custo $c(u, v)$ igual à distância euclidiana entre os pontos u e v.

Mostre que o problema de encontrar uma árvore geradora mínima em um grafo G desse tipo tem cota inferior $\Omega(n \log n)$. Considere o modelo de computação em que só podemos comparar dois números reais por meio de uma caixa-preta, que gasta tempo constante.