Questions from Past Exams

Coping with NP-Hardness

Question 1

Consider the variant of the **Traveling Salesman Problem**, in which the input consists of a three-dimensional array $d_{i,j,k}$, and the cost of traveling from city i to city j depends on the last city k visited just before i. (For the city of origin, 1, the costs of traveling from city 1 to city j are of course independent of k, namely, $d_{1,j,k} = d_{1,j,k'}$ for every $1 \leq k < k' \leq n$.)

Give a dynamic programming algorithm for solving this problem with the best (worst-case) time complexity you can, and prove its correctness and complexity.

Question 2

The **Integer Knapsack** problem is defined as follows:

`Instance:` A collection of n items with integral sizes $a_1, \ldots, a_n > 0$ and profits $p_1, \ldots, p_n > 0$, and an integer $B > 0$.

`Question:` Find integers $x_1, \ldots, x_n \geq 0$ such that $\sum_i a_i x_i \leq B$ and $\sum_i p_i x_i$ is maximized.

Provide a polynomial-time approximation algorithm for this problem with approximation ratio 1/2. Prove the correctness of your algorithm, and establish tight bounds on its approximation quality (i.e., prove that a ratio of 1/2 is always guaranteed, and give an example for an input on which the algorithm fails to do better than that).

Question 3

Consider the variant of the `Traveling Salesman Problem` (TSP), in which the salesman is not required to return to the city of origin at the end of the tour (i.e., the tour is a simple path, rather than a cycle), and it is allowed to start at any of the cities.

Give an algorithm for solving this problem with the best (worst-case) time complexity you can, and prove its correctness and complexity.

Question 4

The `Exact Hitting Set` problem is defined as follows:

Instance: Universe $U = \{1, 2, \ldots, n\}$, collection $C = \{S_1, \ldots, S_m\}$ of subsets $S_i \subseteq U$

Question: Find a minimum size subset $H \subseteq U$ "hitting" each set S_i exactly once (i.e., such that $|S_i \cap H| = 1$ for every i).

Give an algorithm for solving this problem with the best (worst-case) time complexity you can, and prove its correctness and complexity.

Definitions for Questins 5 and 6:

Consider a graph $G(V, E)$.

For vertices $x, y \in V$, let $dist(x, y)$ denote the distance between x and y in G.

For a set Y of vertices in G, let $dist(x, Y) = \min_{y \in Y}\{dist(x, y)\}$.

The *radius* of a set of vertices S in G is defined as $rad(S) = \max_{v \in V}\{dist(v, S)\}$.

Question 5

The r-**Dominating Set** problem is defined as follows:

`Instance:` Graph $G(V, E)$, integer $r \geq 1$

`Question:` Find a minimum size set of vertices S, satisfying $rad(S) \leq r$

Give a polynomial time greedy approximation algorithm for this problem, and analyze its time complexity and approximation ratio.

Question 6

The k-**Centers** problem is defined as follows:

`Instance:` Graph $G(V, E)$, integer $k \geq 1$

`Question:` Find a set S of size $|S| \leq k$, such that $rad(S)$ is minimized

Consider the following greedy algorithm G for the k-Centers problem:

1. $S \leftarrow \{v_1\}$ for some arbitrary vertex v_1.
2. **For** $i = 2$ to k **do**
   (a) Let v be the vertex farthest away from S
   (i.e., such that $d(v, S) \geq d(w, S)$ for every $w \in V$)
   (b) $S \leftarrow S \cup \{v\}$

   end

Analyze the approximation ratio of this algorithm.

Question 7

Define the *volume* of a collection of sets $\{A_1, \ldots, A_k\}$ as $\sum_i |A_i|$.

The `Min Volume Cover` problem is defined as follows:

Instance: Universe $U = \{1, 2, \ldots, n\}$, collection $C = \{S_1, \ldots, S_m\}$ of subsets $S_i \subseteq U$

Question: Find a minimum-volume cover for U

(Recall that a *cover* is a collection of subsets $\{S_{i_1}, \ldots, S_{i_l}\} \subseteq C$ such that $\bigcup_{1 \leq j \leq l} S_{i_j} = U$).

Give a polynomial-time approximation algorithm with the best ratio you can for this problem, and prove your bound.

Question 8

The **Coloring** problem is defined as follows:

`Instance:` Graph $G(V, E)$

`Question:` Find a coloring for the vertices of G with a minimum number of colors, such that no two adjacent vertices have the same color.

Consider the following approximation algorithm for the Coloring problem:

1. Partition the vertices of $G(V, E)$ into $q = n/\log n$ sets $W_1, \ldots, W_q$ of size $\log n$ each. (For simplicity assume both $\log n$ and q are integers.)
2. For each of the sets W_i separately, compute an optimal coloring in the induced subgraph $G(W_i)$, using a different set of colors C_i for each W_i

(a) Give the best bound you can for the approximation ratio of this algorithm, and prove your bound.

(b) Give an example establishing that this bound is tight.

Question 9

Consider the following two decision problems:

Partition:

`Instance:` Integers $a_1, \ldots, a_n > 0$

`Question:` Is there a subset $S \subseteq \{1, \ldots, n\}$ such that $\sum_{i \in S} a_i = \sum_{i \notin S} a_i$?

Sum of Squares:

`Instance:` Integers $a_1, \ldots, a_n, B > 0$

`Question:` Is there a subset $S \subseteq \{1, \ldots, n\}$ such that $(\sum_{i \in S} a_i)^2 + (\sum_{i \notin S} a_i)^2 \leq B$?

Relying on the fact that the Partition problem is NP-complete, prove that the Sum of Squares problem is NP-complete as well.
You may use the following fact without proof: If $x + y = A$ and $x \neq y$, then $x^2 + y^2 > A^2/2$.

Question 10

The K**-largest subset** problem is defined as follows:

`Instance:` Integers $a_1, \ldots, a_n, K, B > 0$

`Question:` Are there at least K distinct subsets $S_1, \ldots, S_K \subseteq \{1, \ldots, n\}$ such that $\sum_{i \in S_j} a_i \geq B$ for every $1 \leq j \leq K$?

Provide a pseudo-polynomial time algorithm for solving the K-largest subset problem. Prove the correctness of your algorithm, and analyze its complexity.

Question 11

Given a graph G and two vertex sets A and B, let $E(A, B)$ denote the set of edges with one endpoint in A and one endpoint in B.

The **Max Equal Cut** problem is defined as follows:

`Instance:` Graph $G(V, E)$, $V = \{1, 2, \ldots, 2n\}$.

`Question:` Find a partition of V into two n-vertex sets A and B, maximizing the size of $E(A, B)$.

Consider the following approximation algorithm for the Max Equal Cut problem:

- Start with empty sets A, B, and perform n iterations.
- In iteration i, pick vertices $2i - 1$ and $2i$, and place one of them in A and the other in B, according to which choice maximizes $|E(A, B)|$.

  (I.e., if $|E(A \cup \{2i-1\}, B \cup \{2i\})| \geq |E(A \cup \{2i\}, B \cup \{2i-1\})|$ then add $2i - 1$ to A and $2i$ to B, else add $2i$ to A and $2i - 1$ to B.)

(a) [15 points] Prove that the algorithm has approximation ratio 2 (i.e., it always finds a partition with at least half the number of edges in the optimal cut).

(b) [10 points] Give an example establishing that this bound is tight.