Describe and analyze (prove correct and give a time analysis) algorithms for any three of the following four problems. Algorithms may be described at "high level" without actual code. You may use any algorithm or data structure from the text or in class, and their correctness and analysis, but be careful. For example, if you construct a graph with \( n^2 \) nodes and \( n^2 \) edges and then run Dijkstra's algorithm on the resulting graph, the total run-time is \( O(n^3 \log n) \), because Dijkstra's algorithm is \( O(\log V) \). If you want to use the correctness of Dijkstra's algorithm on the graph, you must check that edge weights are positive, since otherwise the proof that Dijkstra's algorithm gives shortest paths does not go through.

If a problem has multiple size parameters, you should express the run-time as a function of all relevant parameters; e.g., saying maximum bi-partite matching takes time \( O(|V||E|) \) is more accurate than to say it takes time \( O(V^3) \), although both are correct.

For some problems, correctness may be trivial; for others the analysis may be trivial. If it is trivial, you do not need to go into detail, but you should at least give a one or two line explanation. (Conversely, if you only give a one or two line explanation, I will view this as implicitly claiming that it is trivial.)

Each problem is worth 10 points. Grading may be based on any of the following that are relevant for the problem: the efficiency of your algorithm; the correctness and proof thereof; and the time analysis. You only get credit for what you state and prove: if you give a time analysis of \( O(n^2) \) for your algorithm, and the algorithm is in fact \( O(n \log n) \), your score for efficiency will be based on your claimed \( O(n^2) \). The number of points depending on each part is given after the problem, as well as a ballpark estimate of the time analysis for my solutions of the problem.

**Palindromic path:** You are given a deterministic finite automaton, with alphabet \( \Sigma \) and state space \( V \), specified by its transition diagram, a directed graph where each node \( v \in V \) has exactly one out-edge labelled with each \( \sigma \in \Sigma \), as well as an initial node \( s_0 \) and a set of accepting nodes \( T \subseteq V \). A word \( w \in \Sigma^* \) is accepted by the DFA if the path starting from \( s \) labelled by \( w \) has its endpoint in \( T \). A palindrome is a word that reads the same backwards as forwards. Give an efficient algorithm to tell whether the DFA accepts any palindrome. Analyze its time in terms of \( |V| \) and \( |\Sigma| \). (5 pts correct poly-time algorithm and correctness proof, 5 pts efficiency.)

**Energy Contracts:** UCSD needs long-term contracts for power to keep the lights on. It has a list of \( n \) bids \( B_I, 1 \leq I \leq n \) from power companies, each with a rate (cost per megawatt) \( r_I \) and a capacity \( C_I \) (the maximum
number of megawatts the company can guarantee). The regents, to encourage low bids, has guaranteed that they will pay for all accepted bids at the highest rate of any accepted bid. The accepted bids’ capacities must sum to at least $M$, the university’s demand for power, to ensure enough power. In addition to the money paid per megawatt, the regents expect each contract to cost a fixed amount $F$ for lawyers, setting up connections to the grid, etc. So the total cost will be $Fk + rM$, where $k$ is the number of accepted bids, and $r$ is the maximum rate of an accepted bid. Give an efficient algorithm, polynomial-time in $n$, that, given $F, M$ and the list of $n$ bids $B_I$, computes a subset of bids $A$ that minimizes the cost to the university subject to ensuring enough power. (3 pts correct poly-time algorithm and correctness proof, 7 pts efficiency and time analysis.)

**Regular sub-graph** A directed graph is $d$-regular if every node has exactly $d$ edges entering it and $d$ edges leaving it. The regular sub-graph problem is given a directed graph $G = (V,E)$ and an integer $1 \leq d \leq n$, decide whether there is an edge-induced subgraph $E' \subseteq E$ so that $(V,E')$ is $d$-regular. (For example, when $d = 1$, the question is whether there is a set of disjoint directed cycles covering all of $V$.) Give an efficient algorithm for this problem. (6 pts correct polynomial-time algorithm, 4 pts efficiency and time analysis)

**Approximate min cost clustering** A $[1 - 2]$ metric on $2n$ points is a symmetric distance function $1 \leq d(x,y) = d(y,x) \leq 2$ for each points $x \neq y$. ($d(x,x) = 0$ by convention.) (Note that such a function is always a metric, i.e., always obeys the triangle inequality $d(x,z) \leq d(x,y) + d(y,z)$.) A clustering of the points divides them into two equal size disjoint sets $S$ and $T$ with $|S| = |T| = n$. You want to find a clustering that minimizes the sum of edge weights within a cluster, $\sum_{x,y \in S} d(x,y) + \sum_{x,y \in T} d(x,y)$. Find the best polynomial time approximation algorithm you can for this problem. (Most points based on approximation ratio. Ratio 1.5 is worth most of the points.)