CIS 502 - Algorithms

Fall 2015 Homework $6\frac{6}{7}$ Solutions

Problem 1 *Problem 24, page 331-332, of textbook.*

Solution: Let the number of voters in precinct P_i for the two parties be s_i, t_i respectively (with $s_i + t_i = m$). Assume party 's' has majority overall – therefore the gerrymandering can only help party 's'.

First Solution: We create an array $A[i, k, a, b]$ which stands for: "Is it possible to partition the precincts $P_1, \ldots, P_i$ such that the first part has k precincts and the majority party has a votes in first part and b votes in second part".

Therefore

$$A[i,k,a,b] = \max \begin{cases} A[i-1,k-1,a-s_i,b] & \text{if } P_i \text{ is in the first part} \\ A[i-1,k,a,b-s_i] & \text{if } P_i \text{ is in the second part} \end{cases}$$

Initialization: $A[0,0,0,0] = 1$. Otherwise if $i < k$ or $k > n/2$ then $A[i,k,a,b] = 0$ for any a, b. Otherwise if $i = k \neq 0$ and $b > 0$ then $A[i,k,a,b] = 0$ for any a.

Final answer: Does there exist a, b such that $A[n, n/2, a, b] = 1$ and both $a, b > nm/2$.

The running time is $O(n \cdot (n/2) \cdot (nm) \cdot (nm)) = O(n^4 m^2)$.

The Solution: Observe that $a + b = \sum_{j=1}^{i} s_j$ for any feasible partition of $P_1, \ldots, P_i$. Therefore the coordinate b is superfluous. Thus:

$$A[i,k,a] = \max \begin{cases} A[i-1,k-1,a-s_i] & \text{if } P_i \text{ is in the first part} \\ A[i-1,k,a] & \text{if } P_i \text{ is in the second part} \end{cases}$$

Initialization: $A[0,0,0] = 1$. Otherwise if $i < k$ or $k > n/2$ then $A[i,k,a] = 0$ for any a. Otherwise if $i = k \neq 0$ and $a \neq \sum_{j=1}^{i} s_j$ then $A[i,k,a,b] = 0$.

Final answer: Does there exist a, b such that $A[n, n/2, a] = 1$ and $\sum_{i=1}^{n} s_i - \frac{nm}{2} > a > \frac{nm}{2}$.

The running time is $O(n \cdot (n/2) \cdot (nm)) = O(n^3 m)$.

Problem 2 *Problem 22, page 428, of textbook.*

Solution: An example for part (a) is:

$$\begin{matrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{matrix}$$

For part (b), we construct a bipartite graph (L, R, E) where the set L corresponds to the columns or possible positions of a row, and R corresponds to the rows themselves. Each node of L, R is indicated by i where $1 \leq i \leq n$.

We add an edge (u, v) if $M[v, u] = 1$ (we reversed the role of v, u to preserve the pnemonic of "R" for rows) which means row v can be in position u.

If there exists an arrangement of the rows such that the old row v went to new row $\sigma(v)$ and the column c went to $\pi(c)$ then $M[v, \pi^{-1}(\sigma(v))]$ must have been 1. Now if $\pi^{-1}(\sigma(v)) = \pi^{-1}(\sigma(u))$ for $u \neq v$ then we have $\sigma(u) = \sigma(v)$ (by applying π) and that means that two different rows were

mapped to the same row (and thus a contradiction). Therefore $\pi^{-1}(\sigma(v)) \neq \pi^{-1}(\sigma(u))$ for $u \neq v$. Therefore there exists a matching $(\pi^{-1}(\sigma(v)), v)$ in (L, R, E).

Now, if there exists a perfect matching in (L, R, E), then the matching corresponds to the order in which we arrange the rows. Whichever row v was matched to column/position u must have had $M[v, u] = 1$. After the rearrangement we get $M[u, u] = 1$. This holds for every position u. Note that we only need to permute the rows.

Problem 3 *Problem 27, page 431, of textbook.*

paragraphSolution: Create a bipartite graph where L corresponds to the days and R corresponds to the people. If person j is present in S_i (the set of people going on day i) then draw a directed arc from $i \to j$. Now add a source s which connects to each $i \in L$ with capacity 1 (meaning we need 1 driver for that day) and a sink t which connects to each j with capacity $\lceil \Delta_j \rceil$.

Observe that the fractional fair schedule defines a fractional flow of capacity d from s to t – here 1 unit of flow is sent from s to each i and the flow on the $i \to j$ edge is $\frac{1}{|S_i|}$. The total flow arriving at j is Δ_j and that is less than the capacity $\lceil \Delta_j \rceil$ of the $j \to t$ edge.

Therefore the maxflow is at least d. But there is a cut of capacity (at s) d. Therefore maxflow is d. We can now use Ford Fulkerson to find the maxflow – it will be an integral flow, in time $O(kd^2)$ because the total number of edges is at most kd. The flow can be decomposed into d paths each carrying one unit flow in time $O(kd^2)$. The flow paths through j correspond to the days j is driving. Observe that the schedule is fair by construction. Note this proves the existence, part (a) and the construction, part (b) simultaneously.

Problem 4 *Problem 28, page 519, of textbook.*

Solution: It is easy to see that Strongly Independent Set problem is in NP. Given a set we can easily verify if all the vertices are within distance more than 2.

To prove that it is NP-Hard, we show IS $\leq_m$ SIS. Subdivide each edge to insert a vertex (remove the original edge and add a path of length 2) and add edges between the newly added vertices. If we had an independent set S in the original graph, it is a Strongly Independent set now.

In the reverse direction, if we have a Strongly independent set and $k > 1$ then we cannot have any of the newly added vertices (because all vertices are within length 2 from any of these vertices). Thus the Strongly independent set must contain vertices of the original set – and they must be independent in the original graph (otherwise they will have a path of length 2 in the new graph).