CS-3510-**Your Section#** Algorithms, Spring 2025 Mid-Term 1 Take-Home
Hayden Hawley Collaborator(s):

Problem 1 (10 pts)

For this problem, we explore the issue of *truthfulness* in the Gale-Shapley algorithm for Stable Matching. Show that a participant can improve its outcome by lying about its preferences. Consider $r \in R$. Suppose r prefers p to p', but p and p' are low on r's preference list. Show that it is possible that by switching the order of p and p' on r's preference list, r achieves a better outcome, e.g., is matched with a p'' higher on the preference list than the one if the actual order was used.

Hint: Prove the claim by finding one specific instance of stable matching problem and comparing the stable matching before and after the switching.

Solution

Setup. Let the two sides be:

$$\text{Residents (receivers)} : R = \{r, s, t\}, \quad \text{Programs (proposers)} : P = \{p, p', q\}.$$

We run the program-proposing Gale-Shapley algorithm.

True Preferences

- Resident r : $q \succ p \succ p'$
- Resident s : $p \succ q \succ p'$
- Resident t : $p \succ p' \succ q$
- Program p : $r \succ s \succ t$
- Program p' : $r \succ t \succ s$
- Program q : $s \succ r \succ t$

Case 1: r Reports Truthfully

1. **Round 1.**
   - p proposes to r.
   - p' also proposes to r.
   - q proposes to s.
2. **Resident Decisions.**

- r receives proposals from p and p'. Since r's true order is $q \succ p \succ p'$ (and q did not propose), r prefers p over p'. Thus r holds p and rejects p'.
- s holds q.
- t receives no proposal yet.

3. **Round 2.**
   - p' (rejected by r) proposes to its next choice, t.
   - t accepts p', having no other offer.
4. **Final Matching.**
$$r \leftrightarrow p, \quad s \leftrightarrow q, \quad t \leftrightarrow p'.$$

Here, r ends up with p. Since q went to s, r cannot do better under truthful reporting.

Case 2: r Misreports by Swapping p and p'

Now suppose r reverses p and p' in his submitted list:

$$\text{Reported by } r: \quad q \succ p' \succ p.$$

Then:

1. **Round 1.**
   - p proposes to r.
   - p' also proposes to r.
   - q proposes to s.
2. **Resident Decisions.**
   - r receives proposals from p and p'. According to the reported preference ($p' \succ p$), r holds p' and rejects p.
   - s holds q.
3. **Round 2.**
   - p, rejected by r, proposes next to s.
   - Since s truly prefers $p \succ q$, s drops q and holds p.
   - q, rejected by s, then proposes to r (its next choice).
4. **Round 3.**
   - r is holding p', but now receives a proposal from q.
   - By r's true preference, $q \succ p \succ p'$. So r prefers q and drops p'.
5. **Round 4.**

- p', dropped by r, proposes next to t.
- t accepts p'.

6. **Final Matching.**

$$r \leftrightarrow q, \quad s \leftrightarrow p, \quad t \leftrightarrow p'.$$

In this scenario, r ends up with q, which is strictly better under r's true preference. Thus, by lying about the order of two less preferred programs, r can secure a more-preferred match.

Conclusion

This example shows that when programs propose, a resident (receiver) can sometimes benefit from misreporting: truth-telling is not an optimal strategy for the receiving side.

Problem 2 (10 pts)

Arrange in increasing order of asymptotic growth. All logs are in base 2.

1. $n^{\frac{5}{3}} \log^2 n$
2. $2^{\sqrt{\log n}}$
3. $\sqrt{n^n}$
4. $\frac{n^2}{\log n}$
5. 2^n

Solution

Define the five functions:

$$f_1(n) = n^{\frac{5}{3}} (\log n)^2, \quad f_2(n) = 2^{\sqrt{\log n}}, \quad f_3(n) = \sqrt{n^n} = n^{\frac{n}{2}}, \quad f_4(n) = \frac{n^2}{\log n}, \quad f_5(n) = 2^n.$$

Comparing $f_2(n)$ with polynomials

$$\log_2\left(2^{\sqrt{\log_2 n}}\right) = \sqrt{\log_2 n}, \quad \log_2\left(n^x\right) = x \log_2 n.$$

Clearly, for large n, $x \log_2 n$ outgrows $\sqrt{\log_2 n}$. Hence $2^{\sqrt{\log_2 n}}$ is sub-polynomial. It follows that:

$$f_2(n) \ll n^x \quad \text{and thus} \quad f_2(n) \ll f_1(n),\ f_4(n)$$

since f_1, f_4 are polynomial (up to log factors).

Comparing $f_1(n)$ and $f_4(n)$

$$\frac{f_1(n)}{f_4(n)} = \frac{n^{\frac{5}{3}}(\log n)^2}{\frac{n^2}{\log n}} = \frac{n^{\frac{5}{3}}(\log n)^3}{n^2} = \frac{(\log n)^3}{n^{\frac{1}{3}}}.$$

As $n \to \infty$, $n^{\frac{1}{3}}$ eventually outgrows $(\log n)^3$. Thus

$$f_1(n) \ll f_4(n).$$

Comparing the exponentials $f_3(n)$ and $f_5(n)$

$$f_3(n) = n^{\frac{n}{2}} = \exp\Big(\tfrac{n}{2}\ln n\Big), \quad f_5(n) = 2^n = \exp\big(n \ln 2\big).$$

For large n, $\frac{n}{2}\ln n$ dominates $n \ln 2$. Hence

$$n^{\frac{n}{2}} \gg 2^n.$$

Final Ordering

Putting it all together:

$$f_2(n) < f_1(n) < f_4(n) < f_5(n) < f_3(n).$$

In explicit form:

$$\boxed{2^{\sqrt{\log n}} < n^{\frac{5}{3}}(\log n)^2 < \frac{n^2}{\log n} < 2^n < \sqrt{n^n}.}$$

Problem 3 (10 pts)

We say that $T(n)$ is $O(f(n))$ if there exist c and n_0 such that for all $n > n_0$, $T(n) \leq cf(n)$. Use this definition for parts a and b.

1. Prove that $4n^2 + 3n\log n + 6n + 20\log^2 n + 11$ is $O(n^2)$. (You may use, without proof, the fact that $\log n < n$ for $n \geq 1$.)

2. Suppose that $f(n)$ is $O(r(n))$ and $g(n)$ is $O(s(n))$. Let $h(n) = f(n)g(n)$ and $t(n) = r(n)s(n)$. Prove that $h(n)$ is $O(t(n))$.

Solution

(a) Showing $4n^2 + 3n\log n + 6n + 20(\log n)^2 + 11$ is $O(n^2)$

Let

$$T(n) = 4n^2 + 3n\log n + 6n + 20(\log n)^2 + 11.$$

We check each term for a bound proportional to n^2:

- $4n^2$ is trivially $\leq 4n^2$.

- $3n \log n \le 3n \cdot n = 3n^2$ for $n \ge 1$ (since $\log n \le n$).
- $6n \le 6n^2$ for $n \ge 1$.
- For large n, $\log n \le n$, hence $(\log n)^2 \le n^2$. So $20(\log n)^2 \le 20n^2$.
- $11 \le 11n^2$ for $n \ge 1$.

Summing these bounds shows $T(n) \le (4 + 3 + 6 + 20 + 11)n^2 = 44n^2$ for $n \ge 1$. Therefore, $T(n) \in O(n^2)$.

(b) Product of two $O(\cdot)$ functions

Assume $f(n)$ is $O(r(n))$ and $g(n)$ is $O(s(n))$. Then by definition, there exist constants c_1, c_2 and n_1, n_2 such that for all $n > n_1$, $f(n) \le c_1\, r(n)$, and for all $n > n_2$, $g(n) \le c_2\, s(n)$. Let $n_0 = \max\{n_1, n_2\}$. Then for $n > n_0$,

$$f(n)\, g(n) \;\le\; c_1\, r(n) \;\cdot\; c_2\, s(n) \;=\; (c_1 c_2)\, \big(r(n)\, s(n)\big).$$

Hence $f(n)\, g(n) \in O\big(r(n)\, s(n)\big)$.

Problem 4 (10 pts)

The *diameter* of an undirected graph is the maximum distance between any pair of vertices. If a graph is not connected, its diameter is infinite. Let G be an n node undirected graph, where n is even. Suppose that every vertex has degree at least n/2. Prove that G has diameter at most 2.

Hint: Proof by contradiction

Solution

Let G be an undirected graph on n vertices, with n even, and every vertex of degree at least $\frac{n}{2}$. We will show the diameter of G cannot exceed 2.

Step 1: Contradiction assumption. Assume $\mathrm{diam}(G) \ge 3$. Then there exist vertices u, v such that the distance $d(u, v) \ge 3$. In particular:

1. u and v are not adjacent (otherwise $d(u, v) = 1$),
2. There is no single vertex adjacent to both u and v (otherwise $d(u, v) = 2$).

Step 2: Neighborhood sizes. Let $N(u)$ be the neighbors of u and $N(v)$ be the neighbors of v. Since each vertex has degree at least $\frac{n}{2}$,

$$|N(u)| \;\ge\; \frac{n}{2}, \quad |N(v)| \;\ge\; \frac{n}{2}.$$

Because no vertex is adjacent to both u and v, the sets $N(u)$ and $N(v)$ must be disjoint. Hence

$$|N(u) \cup N(v)| = |N(u)| + |N(v)| \geq \frac{n}{2} + \frac{n}{2} = n.$$

But there are only $n-2$ vertices in total other than u and v, so it is impossible for two disjoint neighbor sets to each contain $\frac{n}{2}$ different vertices. We have a contradiction.

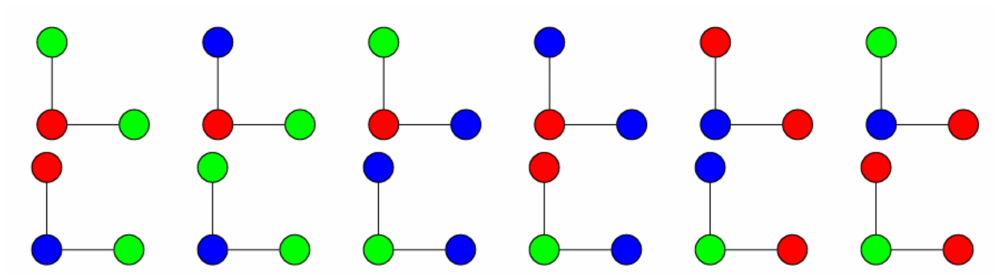
Step 3: Conclude diameter at most 2. Since assuming $\mathrm{diam}(G) \geq 3$ led to a contradiction, it follows that every pair of vertices in G is at distance at most 2. Thus the diameter is ≤ 2.

$$\boxed{\text{Hence, any } n\text{-vertex graph with minimum degree } \tfrac{n}{2} \text{ has diameter at most 2.}}$$

Problem 5 (10 pts)

Show that there are at least $3 \cdot 2^{n-1}$ ways to properly color vertices of a tree T with n vertices using 3 colors, i.e., to color vertices with three colors such that any two adjacent vertices have distinct colors. Note that it can be shown that there are exactly $3 \cdot 2^{n-1}$ ways to properly color vertices of T with 3 colors but in this problem, to receive full credit, it is enough prove the "at least" part.

For example, there are (at least) $3 \cdot 2^2 = 12$ ways to color a tree with 3 vertices as show below:



Problem 6 (Extra Credit: 10 pts)

Given a directed graph G with n vertices $V = \{1, 2, \cdots, n\}$ and m edges. We say that a vertex j is reachable from i if there is a directed path from i to j. Design an $O(m+n)$-time algorithm (show the pseudo-code) that for any vertex i outputs the smallest label reachable from i. For example, given the following graph you should output 1,2,2,2,1 corresponding to the smallest indices reachable from vertices 1,2,3,4,5 respectively.