CS 5800: Algorithms - Assignment-1

Ajeya Kempegowda

January 27, 2019

1. Consider an instance of the Stable Matching Problem in which there are n employers and n job applicants. Each employer has exactly one job offering, and each applicant applies to all n available jobs. Suppose there exists an employer e and an applicant a such that e is ranked first on the preference list of a and a is ranked first on the preference list of e. Then in every stable matching S for this instance, the pair (e, a) belongs to S.

   Decide whether you think the statement above is true or false.

   True.

   Consider the situation where group 1, e2, a1, and a2.

   e1 has the ranking: (a1, a2)

   e2 has the ranking: (a2, a1)

   a1 has the ranking: (e1, e2)

   a2 has the ranking: (e2, e1)

   Consider each applicant, a1 and a2:

   If $a1$ were in a pair with the $e's$ ranked first on the preference list of $a1$, then $a1$ would be in a pair with $e1$ and similarly, $e2$ would be in a pair with $a2$, whom is ranked first on $e2's$ preference list, which directly follows Gale-Shapley Algorithm.

   Suppose if there's situation where $e1, e2$ have the same preference list-(a1,a2), there would always be a stable matching pairs(e,a) such that (e1,a1),(e2,a2) ϵ S or (e2,a1),(e1,a2) ϵ S.

   Therefore, in this situation, it's possible to have a pair (e, a), let alone a stable matching containing a pair, such that a is ranked first on the preference list of a and a is ranked first on the preference list of e.

2. **Stable Matching with Unequal Sets**

   INITIALIZE all m ϵ H and n ϵ R are free.

   WHILE some hospital H is unmatched(free) for all positions and has not proposed to every resident such that (m,n)$\notin M$

   m1 $\leftarrow$ be such an hospital

   r1 $\leftarrow$ first student on H's list to whom H has not yet proposed

IF (r1 is unmatched)

Add (m1–r1) to matching M.

ELSE IF (r1 is currently engaged to m2 **and** r1 prefers m1 to m2)

Replace (r1–m2) in matching M with (r1–m1)

Release m2 back to the unmatched pool

ELSE

r1 rejects m1.

Return the set M of matched pairs

Proofs for stable matches from the above algorithm:

- ***Hospitals propose to residents in decreasing order of preference***. The above algorithm inherently takes care of the first type of instability stated. If h prefers $s\,'$ to s, then $s\,'$ would be ranked higher than s in $h's$ preference list and as aforementioned, the algorithm would propose to residents based on order of preference. Therefore, the algorithm handles this situation effectively.
- **Once a resident is matched, they can never be unmatched; they only trade up** - *which takes care of second instability stated.* If there's a situation where h prefers $s\,'$ to s, the current engagement(h, s) would be replaced by $(h, s\,')$ and s would be released to the unmatched pool and the algorithm repeats. The same situation can be used to explain the counterpart where s prefers $h\,'$ to h.
- Algorithm terminates after at most $(m * n)$ iterations of while loop.

3. **Proof by Cases**

Let P be the assertion that w is even; then w^2 *is also even and always be a multiple of 4*. We would be using these two axioms to prove the assertions

Let R be the assertion that that all three of x, y, z are even.

Given P, there are four cases we have to account for:

- x, y, z are odd integers.
- x is odd; y, z are even.
- x, y are odd integers; z is even.
- x, y, z are even.

Given that,

$$x^2 + y^2 + z^2 = w^2 \tag{1}$$

- **Case 1**:

  x, y, z are odd integers, such that x, y, z can be represented as 2m+1; m $\forall \epsilon\ z^+$

  Then,

$$w^2 = x^2 + y^2 + z^2 = 4(X^2 + Y^2 + Z^2 + Z + Y + X) + 3 \tag{2}$$

  Clearly, the above equation cannot represent a even number, as an odd number added to a multiple of 4 cannot yield an even number. Hence, x, y, z cannot be odd integers.

- **Case 2**:

  Let, x and y be odd integers and z represent an even integer such that z = 2m; m $\forall\epsilon$ z^+
  Then,
$$w^2 = x^2 + y^2 + z^2 = 4(X^2 + Y^2 + Z^2 + X + Y) + 2 \tag{3}$$

  The above equation yields an even integer for all integer values of x, y, z but they need not be necessary a multiple of four. For example for x=3, y=5 and z=2
$$3^2 + 5^2 + 2^2 = 9 + 25 + 4 = 38 \tag{4}$$
  which is even but not a multiple of 4. Hence, even this case fails, meaning that P does not hold true for all values.

- **Case 3**:

  Let, x be an odd integer and y, z represent even integers.
  Then,
$$w^2 = x^2 + y^2 + z^2 = 4(X^2 + Y^2 + Z^2 + X) + 1 \tag{5}$$

  The above equation follows case 1. The gist of the above equation can be boiled down to (multiple of 4) + 1. Which, clearly cannot be an even integer. Therefore, even this case fails.

- **Case 4**:

  Let, x, y, z represent even integers such that that x, y, z = 2m; m $\forall\epsilon$ z^+
  Then,
$$w^2 = x^2 + y^2 + z^2 = 4(X^2 + Y^2 + Z^2) \tag{6}$$

  The above equation is a multiple of 4 and for all integer values of x, y, z. Clearly, the result would be a multiple of 4 and would be even. In other words,
$$P \Rightarrow R \tag{7}$$
  Conversely, to prove $R \Rightarrow P$, where R asserts that x, y, z are even integers,
$$x^2 + y^2 + z^2 = 4(X^2 + Y^2 + Z^2) \tag{8}$$
  The right hand side is a multiple of 4. Any number when multiplied by an even number yields another even number and for any positive integer values of x, y, z the resulting value would be divisible by 4 . Hence the assertion holds true. Therefore, we can conclude that,
$$P \Leftrightarrow R \tag{9}$$

4. **Asymptotic Growth Rate**

   (a) **f(n) =** $n2^n$ **and g(n) =**3^n

   For convenience let us choose the following:
   $g(n)/f(n) = 3^n/n2^n$
   $= 1/n(3/2)^n$
   $= 1/n(1.5)^n$
   Applying L Hopital's rule:
   $= \lim_{n\to\infty} ln(n)(1.5)^n/1$
   Applying limits yields to ∞
   This mean that $g(n)$ is growing faster than $f(n)$. Therefore we can say, **f(n) =** O **g(n)**

(b) **f(n) =** $n(log_3 n)^5$ **and g(n) =**$n^{1.2}$

$g(n)$ can be re-written as:

$n^{1.2} = n^{6/5} = (n * n^{1/5})$

$g(n)/f(n) = n * n^{1/5}/n(log_3 n)^5$

$g(n)/f(n) = n^{1/5}/(log_3 n)^5$

Applying L Hopital's rule and differentiating numerator and denominator:

$g(n)\,'/f(n)\,' = (1/5)(n^{-4/5})/(5(log_3 n)^4 * 1/nln(3))$

moving 1/5 and ln(3) to the numerator and further simplifying

$g(n)\,'/f(n)\,' = (1/25)/(log_3 n)^4 * (n^{(-4/5)+1} * ln(3))$

$g(n)\,'/f(n)\,' = (1/5)^2 * ln(3) * n^{1/5}/(log_3 n)^4$

$ln(3)/25$ is a constant which can be retained. The rest of the expression looks similar to the expression we started off with. There we can generalize the process.

If we apply the L Hopital's rule again, we would get result as:

$g(n)\,''/f(n)\,'' = (1/5)^3 * (1/4) * (ln(3)^2) * n^{1/5}/(log_3 n)^3$

Further,

$g(n)\,'''/f(n)\,''' = (1/5)^4 * (1/4) * (1/3) * (ln(3)^3) * n^{1/5}/(log_3 n)^2$

We can boil down the final result to,

$= (1/5)^6 * (1/4)(1/3)(1/2) * (ln(3)^5) * n^{1/5}$

$= (1/5)^5 * (1/5)(1/4)(1/3)(1/2) * (ln(3)^5) * n^{1/5}$

$= (1/5)^5 * (1/5!) * (ln(3)^5) * n^{1/5}$

Evaluating the limits yields the value ∞

This mean that $g(n)$ is growing faster than $f(n)$. Therefore we can say, **f(n) =** O **g(n)**

(c) **f(n) =** $2^{\sqrt{l}ogn}$ **and g(n) =**n^{logn}

Let, $z = \sqrt{l}ogn$, then $z^2 \Rightarrow logn$

f(n) becomes 2^z and g(n) $= n^{z^2}$.

By the properties of logarithms n can re re-written as $n = e^{z^2}$

g(n) becomes: $e^{z^2 z^2} = e^{z^4}$

Comparing the bases of $f(n) = 2^z$ and $g(n) = e^{Z^4}$ we can conclude that g(n) > f(n) as $e = 2.71$

This mean that $g(n)$ is growing faster than $f(n)$. Therefore we can say, **f(n) =** O **g(n)**

5. **Asymptotic Functions**

Given, f(n)= θ (r(n)) and g(n) = θ (s(n))

By definition:

$$\theta(r(n)) = 0 \leq C_1 * r(n) \leq f(n) \leq C_2 * r(n) \tag{10}$$

$$\theta(s(n)) = 0 \leq C_1\,' * s(n) \leq g(n) \leq C_2\,' * s(n) \tag{11}$$

Multiplying these two equations:

$$\theta(r(n)) * \theta(s(n)) = 0 \leq (C_1 * C_1\,') * (r(n) * s(n)) \leq (f(n) * g(n)) \leq (C_2 * C_2\,') * (s(n) * r(n)) \tag{12}$$

Further simplification of the equations,

$$r(s) * s(n) = t(n); \tag{13}$$

$$f(n) * g(n) = h(n); \tag{14}$$

$$C_1 * C_1{}' = K_1; \tag{15}$$

$$C_2 * C_2{}' = K_2; \tag{16}$$

where, h(n) and t(n) are functions obtained by the product of $f(n) * g(n)$ and $r(s) * s(n)$ respectively. Similarly, K_1 and K_2 are constants ≥ 0 obtained by the product of $C_1 * C_1{}'$ and $C_2 * C_2{}'$

Now, (12) can be re-written as:

$$h(n) = 0 \leq K_1{}' * t(n) \leq h(n) \leq K_2{}' * t(n) \tag{17}$$

Which is equivalent to the form of

$$h(n) = \theta(t(n)) \tag{18}$$