Homework 2

Try to be precise and to the point. Your answers should be short.

RUNNING TIMES:

1. Consider Euclidean algorithm. The input integers n and m are given in their decimal representations.
   (a) What is the size of the input for the algorithm?
   (b) Explain why the running time of the algorithm is linear on the size of the input. For the analysis, assume that the division process takes a constant amount of time.

   Solution:
   (a) $1(a)$: the length of the decimal representation is the size of the integer. The input size is $\max\{size(n), size(m)\}$ or $size(n) + size(m)$.
   (b) $1(b)$: With each iteration the max value of a, b is at least twice less than the max value of a and b before the iteration. This implies that there are $O(size(input))$ iterations of the while loop. Each iteration takes a constant time. Hence the algorithm runs in linear time.

2. Let p be a polynomial with positive coefficients. Show that p is $O(n^{\log(n)})$.

   Solution: From the lectures we can assume that $p(n) = n^k$ for some fixed k. Then for all $n > 2^k$ we have $n^{\log(n)} \geq n^{\log 2^k} = n^k$. Hence, $O(n^{\log(n)})$.

3. Show that $\sum_{i=1}^{n} i^2 = \Theta(n^3)$.

   Solution: We have:

   $$\sum_{i=1}^{n} i^2 = 1 + 2^2 + 3^2 + \ldots + n^2 \leq n^2 + n^2 + \ldots + n^2 = n^3.$$

   So, $\sum_{i=1}^{n} i^2 = O(n^3)$. On the other hand, we have

   $$\sum_{i=1}^{n} i^2 > (n/2)^2 + (n/2+1)^2 + \ldots n^2 > (n/2)^2 + (n/2)^2 + \ldots (n/2)^2 = (n/2)(n/2)^2 = (1/8)n^3.$$

   So, we have $n^3 = O(\sum_{i=1}^{n} i^2)$.

4. Show that $n \log(n) = \Omega(\log(n!))$.

   Solution: Note that

   $$n! = 1 \cdot 2 \cdot 3 \cdot \ldots \cdot n < n \cdot n \cdot n \cdot \ldots \cdot n = n^n.$$

   Hence,

   $$\log(n!) < \log(n^n) = n \log(n).$$

5. Show that $a^n = O(n!)$ for any positive integer $a > 1$.

   Solution: For simplicity assume that a is an integer. Note that

   $$a^n = a \cdot a \cdot \ldots \cdot a = a^a \cdot a^{n-a} < a^a(a!)(a+1)\cdot(a+2)\cdot\cdot(n-1)\cdot n$$

   Hence,

   $$a^n < Cn!,$$

   where C is the constant a^a. Hence $a^n = O(n!)$.

6. Show that

   $$\sum_{i=1}^{n} \log(n/i) = \Theta(n).$$

   Show this directly. For instance, do not use some advanced results from calculus such as Stirling formula.

   Solution: This is a bit tricky one. Note this first:

   $$\sum_{i=1}^{n} \log(n/i) = \sum_{i=1}^{n}(\log(n) - \log(i)) = n\log(n) - \sum_{i=1}^{n}\log(i)$$

   Let us denote this sum by s_n. Then an easy arithmetic gives us:

   $$s_{n+1} - s_n = (n+1)\log(n+1) - n\log(n) - \log(n+1) = n\log(1+1/n).$$

   When n becomes bigger and bigger the value $\log(1+1/n)$ can be approximated by $1+1/n$. Hence, the value

   $$s_{n+1} - s_n$$

   asymptotically equals 1. Hence,

   $$s_n = (s_n - s_{n-1}) + (s_{n-1} - s_{n-2}) + (s_{n-2} - s_{n-3}) + (s_{n-3} - s_{n-4}) + \ldots + (s_2 - s_1) + s_1 = O(n).$$

7. For the codes below analyse the running times in terms of Θ notation.

Algorithm 1

For $i = 1$ to n do
$j = n - i$
while $j \geq 0$ do
$j = j - 3$.

Solutions: The outer loop has n iterations. When $i = 1$, the inner loop has about $(n-1)/3$ iterations. When $i = 2$, the inner loop has about $(n-2)/3$ iterations. When $i = 3$, the inner loop has $(n-3)/3$ iterations. So the total number of iterations is the sum

$$\sum_{i=1}^{n}(n-i)/3 = \Theta(n^2).$$

Algorithm 2

Set $s = 0$
for $i = 1$ to n do
for $j = 3 \cdot i$ to n do
$s = s + 1$.

Solution: When $i = 1$, the inner loop has about $n - 3$ iterations. When $i = 2$, the inner loop has $n - 6$ iterations. When $i = 3$, the inner loop has $n - 9$ iterations. So the total number of iterations is the sum

$$(n-3) + (n-6) + (n-9) + \ldots = \Theta(n^2).$$

Algorithm 3

for $i = 1$ to n do
$j = i$
while $j < n$ do
$j = 2 \cdot j$.

Solution: When $i = 1$, the inner loop has about $\log(n)$ iterations. When $i = k$, the inner loop has $\log(n/k)$ iterations. So the total number of iterations is the sum

$$\sum_{i=1}^{n} \log(n/i) = \Theta(n)$$

as we alrready proved above.

GRAPHS:

1. Let G be any graph. Explain why the sum of all degrees of the vertices of any graph G equals twice the number of edges of G.

   Solution: Consider the sum of the degrees of G:

   $$\sum_{v \in V} deg(v).$$

   In this sum every edge $\{u, v\}$ contributes 1 to the degree of v and the degree of u. Hence, the sum equals twice the number of edges. Here use only marks 0 or 2 (dont give 1)

2. How many edges does a complete bipartite graph $K_{n,m}$ have?

   Solution: $n \cdot m$.

3. Prove that if there is a path in a graph from vertex x to vertex y and $x \neq y$ then there is a simple path from x to y.

   Solution: Let $x = v_0, v_1, \ldots, v_n = y$ be a path from x to y. If no vertices in this path are repeated then the path is simple and we are done. Otherwise, there are two positions $i < j$ in the path such that $v_i = v_j$. In the original path

   $$x = v_0, v_1, \ldots, v_n = y$$

   remove the subpath $v_{i+1}, \ldots, v_j$. By removing this subpath we obtain the sequence:

   $$x = v_0, \ldots, v_i, v_{j+1}, \ldots, v_n.$$

   This is still a path from x to y and its length is strictly less than the length of the original path. If this new path is simple then we are done. Otherwise, repeat the process. This process produces a simple path from x to y.

4. Show that every finite connected graph G with more than 1 vertex has two vertices of the same degree.

   Solution: Say G has n vertices. Each vertex v has at most $n-1$ edges. So, the degree of each vertex is either 0 or 1 or 2 or $\ldots$ $n-1$. Since G is connected no vertex has degree 0. Hence, the degree of each vertex is either 1 or 2 or $\ldots$ $n-1$. Since there are n vertices on G, at least two of them must have the same degree.

5. We call a graph 3-regular if every vertex of the graph has degree 3. Draw 3-regular connected graphs consisting of 4 vertices, 6 vertices, and 8 vertices.

   Solution: One needs just to draw such graphs. It is explained in the lecture or tutorial.

6. Let G be a connected graph. For any two vertices u, v let $d(u, v)$ be the distance from u to v. Recall that the distance from u to v is the length of the shortest path from u to v. Prove the following triangle inequality. For all vertices x, y and z of the graph we have

   $$d(x, z) \leq d(x, y) + d(y, z).$$

   Solution: Let $P(a, b)$ be the shortest path from a to b. Concatinating the paths $P(x, y)$ and $P(y, z)$ we get the path P from x to z. Note that the length of P is not shorter than the length of $P(x, z)$. Therefore we have

   $$d(x, z) = length\ P(x, z) \leq length\ P = length\ P(x, y) + length\ P(y, z) = d(x, y) + d(y, z).$$