CS131 Homework #8 (17 pts)

1) (6 pts) Consider the three algorithms for computing a^n (n – natural number)

Algorithm 1: iterative (a^0=1, if n>0 multiply a by itself n times)

Algorithm 2: recursive

power(a: nonzero real number, n: nonnegative integer)

if $n = 0$ then return 1

else return $a\cdot$ *power* ($a, n - 1$)

Algorithm 3: recursive, divide-and-conquer, based on $a^n= (a^{n/2})^2$ if n is even; $a^n= a(a^{\lfloor n/2 \rfloor})^2$ if n is odd

a. (2 pts) For the recursive algorithms write recurrence relations describing number of operations f(n) for natural n for algorithm 2 and even n for algorithm 3. (Hint: for algorithm 2 the recurrence relation is linear non-homogeneous of 1st degree: f(n)=…f(n-1)+…; for algorithm 3 the recurrence relation is of divide-and-conquer form: f(n)=…f(n/…)+…)

Algorithm 2: f(n)=f(n-1)+1, f(0)=0

Algorithm 3: f(n)=f(n/2)+1, f(0)=0 (you compute $(a^{n/2})^2$ in f(n/2) operations, then compute $(a^{n/2})^2$ as $(a^{n/2})\cdot(a^{n/2})$.

b. (3 pts) Estimate O(g(n)) complexity of each of the three algorithms. (Hint: for algorithm 2 solve as nonhom. lin. rec. relation as in slides 34-37 of Lecture 11 or using backward substitution as in slide 10 of Lecture 11; for algorithm 3 use Master theorem)

Algorithm 1: n multiplications: O(n)

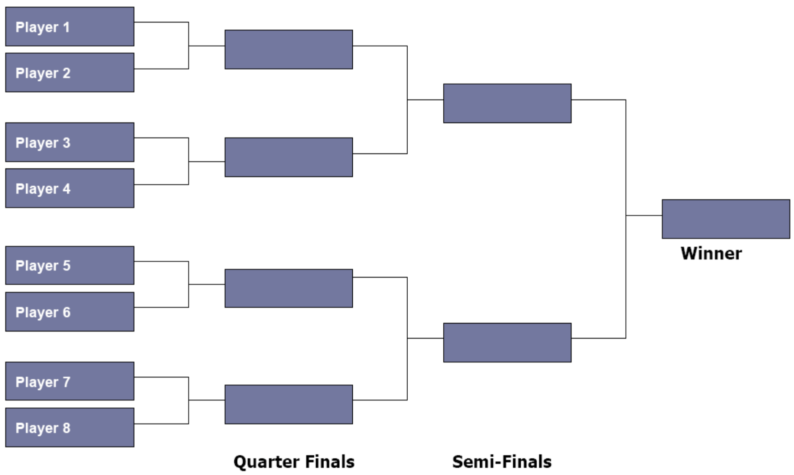
Algorithm 2: f(n)=f(n-1)+1= f(n-2)+2=f(n-k)+k=f(0)+n=n. f(n)∈O(n)

Algorithm 3: f(n)=f(n/2)+1. By Theorem 1 from Lecture 12 (or by using backward substitutions) f(n)∈O(log n)

c. (1 pt) Which one is the fastest?

Algorithm 3 is the fastest (for big n).

2) (5 pts) Suppose that there are $n = 2^k$ teams in an elimination tournament, where there are $n/2$ games in the first round, with $n/2$ winners playing in the second round, and so on.



a. (2 pt) Develop a recurrence relation for $f(n)$ – the number of games in the tournament.
   $f(n) = f\left(\frac{n}{2}\right) + \frac{n}{2}, f(1) = 0$

b. (1 pt) Give a big-O estimate of $f(n)$, explaining your result.
   Applying Master theorem with $\mathrm{a} = 1, \mathrm{b} = 2, \mathrm{d} = 1, \mathrm{a} < b^d$, we get f(n)∈O(n).
   Solve the recurrence relation via the method of backward substitution, get the formula for $f(n)$ via n.
   $f(n) = f\left(\frac{n}{2}\right) + \frac{n}{2} = f\left(\frac{n}{4}\right) + \frac{n}{2} + \frac{n}{4} = \cdots$

$$= f\left(\frac{n}{2^k}\right) + \mathrm{n}\left(\frac{1}{2} + \frac{1}{4} + \cdots + \frac{1}{2^k}\right) = f(1) + 2^k\left(\frac{\frac{1}{2}\left(1 - \frac{1}{2^k}\right)}{1 - \frac{1}{2}}\right)$$

$$= f(1) + 2^k\left(1 - \frac{1}{2^k}\right) = 2^k - 1 = n - 1.$$

c. (1 pt) Check whether your formula matches the result obtained in b. (If it does not – fix the errors.)
   f(n)=(n-1)∈O(n) - correct

d. (1 pt) The tournament diagram for 8 players is shown above. Make tournament diagrams n=2, 4. Check whether your formula correctly counts the number of games when the number of teams n=2, 4, 8. (The number of games produced by the formula should be equal to the number of games shown on the tournament diagrams; if it doesn't – fix the errors.)
   2 teams - 1 game (to determine the winner among two) f(2)=2-1=1
   4 teams - 3 games f(4)=4-1=3
   8 teams - 7 games (see the diagram above) f(8)=8-1=7

3) (4 pts) There are four possibilities for each base in DNA: A, C, G, and T. How many 5-element DNA sequences of bases

   a. end with A?

   $4^4 = 256$

   b. start with T and end with G?

   $4^3 = 64$

   c. contain only A and T?

   $2^5 = 32$

   d. do not contain C?

   $3^5 = 243$

   Grading comments: correct left-hand side (3^5) or right-hand side (243) is enough to get full credit.

4) (1 pt) A committee is formed consisting of one representative from each of the 50 states in the US, where a representative from a state is either the governor or one of the 2 senators from that state. How many ways are there to form this committee?

   3^{50}= **717,897,987,691,852,588,770,249**

   (Grading comments: if a student just writes 3^{50} - it is ok.)

5) (1 pt) How many license plates can be made using either 3 uppercase English letters followed by 3 digits or 4 uppercase English letters followed by 2 digits?

   $26^3 10^3 + 26^4 10^2 =$ **63,273,600**

   (Grading comments: if a student just writes $26^3 10^3 + 26^4 10^2$ - it is ok.)