REC 14: Algorithms (optional, no credit)

Problem 1 Problem 5 Play Poker: Texas Hold'em

Form 3-5 tables of 4-6 students each. Purpose: obtain the highest value 5-card combination using any of the 2 "hole" cards and any of the "community" 5 cards. See picture for combinations ranked from high to low.

Chips value
\$1=White \$5=Red \$25=Green \$100=Black

One of you can act as the dealer. Game rules:

- everyone is dealt 2 "hole" cards. After looking at the cards, each player in clockwise order can decide to play (\$1 fee) or to fold. The remaining players can do the first round of betting: in clockwise order, either
  - "check" : don't change the pot, assuming no raise happened
  - "call" : match highest bet
  - "raise" : increase the the highest bet
  - "fold" : give up and lose the current pot.

  There can be at most 2 raises of \$1 each for a total pot of \$3/player.
- The dealer reveals face up the "flop" 3 community cards. Players update their chances of wining, and another round of betting follows up. At most three raises of at most \$3 each to a pot of max \$12/player.
- The dealer reveals face-up the "turn" a forth community card. Another round of betting of at most 3 raises of up to \$3 each can take the pot to max \$21/player
- The dealer reveals face-up the "river" a fifth and final community card. Last round of betting can be at most 3 raises of at most \$5 each take the pot to max \$36 /player.

Problem 2

Recall that $f(x) = O(g(x))$ means that:

$$\exists c, x_0 \in \mathbb{R} \text{ with } x_0 \leq x \rightarrow 0 \leq f(x) \leq cg(x)$$

Using this definition, show each of the statements below is true or explain, in one sentence, why the statement is false. To show one of these statements is true, write out the conditional above with your values for c, x_0 plugged in.

i $10x = O(x^3)$

ii $4x^2 - x + 2 = O(x^2)$

iii $x! = O(\log_2 x)$

Problem 3

As a developer for a new turn-by-turn navigation app, you have built two algorithms which determine the best path between two positions on the map. If there are n total inersections on the map then method 0 uses:

$$T_0(n) = 100n \log_2 n \text{ computations}$$

while method 1 uses:

$$T_1(n) = n^2 \text{ computations}$$

i If there are 10 intersections on the map, which method uses fewer computations?

ii If there are 2^{15} intersections on the map, which method uses fewer computations?

iii Which method grows slower (under the big-O definition of function growth you can just lookup their corresponding "buckets")? Explain the meaning of slower function growth so its benefit is easily understood.

iv What is the smallest number of intersections n (whole number) which ensures that method 0 is faster (fewer computations) than method 1? Note that the equation you generate may not be easily solved with pencil and paper, use Wolfram Alpha https://www.wolframalpha.com/input?i=x%5E2+%2B+5x+%2B+6+%3D+17+x to compute a final answer (the "solutions" box on the linked page may be helpful).

Problem 4

Identify the function growth bucket of each of the following functions. (Remember, you may discard multiplying coefficients in each term as well as all slower growing functions to find the proper bucket)

i $100n^3 + 2n + 7$

ii $123 + n\log_2 n + 999n$

Problem 5

Solve for x in each of the equalities below:

i $\log_2 16 = x$

ii $\log_x 125 = 3$

iii $\log_2 x = 7$

iv $\log_x 625 - \log_x 25 = 2$

Problem 6

Solve each of the following recurrences by substitution. Assume a base case of $T(1) = 1$. As part of your solution, you will need to establish a pattern for what the recurrence looks like after the k-th substitution. Check that this pattern is consistent with your substitutions, but you do not need to formally prove it is correct via induction.

i $T(n) = T(n-1) + 1$

ii $T(n) = T(n-3) + 4$