CS3230, Mid Term

Answer all questions.

1. We claim that $T(n) = \Theta(n^2)$.

   (a) We show by induction that $T(n) \leq n^2 + 1$:

   For $n = 0, 1, 2, 3$, it clearly holds.

   Suppose it holds for $n \leq k$.

   Then, for $n = k + 1$,

   $T(k+1) = T(k-2) + k + 1 \leq (k-2)^2 + 1 + k + 1 \leq (k+1)^2 + 1$, as $k \geq 3$.

   (b) We show by induction that $T(n) \geq n^2/100$.

   This clearly holds for $n = 0, 1, 2, 3$.

   Suppose it holds for $n = k$.

   Then, for $n = k + 1$

   $T(k+1) = T(k-2) + k + 1 \geq \frac{(k-2)^2}{100} + k + 1 = \frac{k^2-4k+4+100k+100}{100} \geq (k+1)^2/100$.

2. Not optimal, as for $S = 15$, optimal gives three 5's, though the greedy algorithm gives one 12 and three 1's.

A	B	C	D	E	F	Rem
0	∞	∞	∞	∞	∞	A, B, C, D, E, F
0	2	∞	5	10	∞	B, C, D, E, F
0	2	5	5	8	∞	C, D, E, F
0	2	5	5	8	10	D, E, F
0	2	5	5	7	10	E, F
0	2	5	5	7	9	F
0	2	5	5	7	9	

Table 1: Dist Array and Rem

3. The last row above gives the distance from A to different vertices at the end of the Dijkstra's algorithm.

4. By induction on n.

   For $n = 1$, it was shown in class.

   Suppose by induction that $4 \times (3k+1)$ board with a missing square can be tiled using trominoes.

   Then, for $n = k + 1$, consider the board of size $4 \times (3k+4)$ with a missing square.

   The missing square is either in the first $3k$ columns or in the last 4 columns.

   If the missing square is in the first $3k$ columns, then by induction one can tile the $4 \times (3k+1)$ part with a missing square; the 4×3 part can be tiled using tutorial 4 Q4 method.

   On the other hand, if the missing square is in the last 4 columns, then the $4 \times 3k$ board can be tiled using Tutorial 4 Q4 method, while the 4×4 board with missing square can be tiled using the method done in lecture.

5. Without loss of generality assume that all d_i are > 0, and the end point of journey is larger than all d_i (otherwise, we can ignore the $d_i \leq 0$, and the $d_i \geq$ end point of journey).

   Assume that initially the tank is full.

   Let $d_0 = 0$, and d_{n+1} = endpoint for the following.

   Assume that $d_{i+1} - d_i \leq D$ in above (otherwise the journey cannot be performed).

   1. Assume that the d_i's are in sorted order (otherwise sort them).
   2. $laststop = 0$.
   3. $stops = \emptyset$.
   3. For $i = 1$ to n do {
    If $d_{i+1} - d_{laststop} > D$, then {
        $stops = stops \cup \{i\}$
        $laststop = i$ }
}
   4. End.

   To show that the above algorithm gives optimal number of stops, suppose the stops used by the above alogrithm are $A_1, A_2, \ldots A_k$. Suppose the optimal algorithm uses stops $B_1, B_2, \ldots, B_r$ and $r \leq k$. Let $A_0 = B_0 = 0$.

   We first claim by induction that $A_j \geq B_j$, for $0 \leq j \leq r$.

   This is clearly true for $j = 0$ (by definition).

   Now suppose by induction that $A_w \geq B_w$. Then, as $d_{A_{w+1}+1} - d_{A_w} > D$, and $B_w \leq A_w$, we have that $B_{w+1} \leq A_{w+1}$ (otherwise the optimal method doesn't have enough fuel to go from stop B_w to stop B_{w+1}).

   It follows that $B_r \leq A_r$.

   As optimal is able to reach the destination, this implies that $d_{n+1} - d_{B_r} \leq D$. But, then $d_{n+1} - d_{A_r} \leq D$, and the algorithm would not fill gas any more beyond the stoppage at A_r. Thus $r = k$.