EN. 520.650 HW1

1. a) Playing soccer: Performance measure: sport, team, goal,
Environment: lawn, ~~weather~~ weather
Actuators: football, soccer player, goal gate
Sensors: goal sensor, monitor camera, ~~a~~ referee

b) Knitting a sweater: Performance measure: speed, quality, color
Environment: home, ~~tools~~, sweater thread
Actuators: weaver, tools
sensor: timer, quality judge

c) Bidding on an item at an auction: performance measure: quality, price
Environment: auction
Actuator: auctioneer, customer
sensor: timer, price counter

Task Environment	Observable	Agents	Deterministic	Episodic	Static	Discrete
Playing soccer	fully	Multi	strategic	episodic	Dynamic	discrete
knitting a sweater	fully	single	deterministic	sequential	Dynamic	continuous
Bidding ---auction	partially	Multi	strategic	episodic	Dynamic	discrete

2.
- Supermarket bar code scanners: no need the artificial intelligence,
- Web search ~~agent~~ engines: high extent, such as some search recommed algorithms.
- Voice-activated telephone menus: It need accurate detect of voice and high intelligence response, it need high extent AI
- Spelling and grammar correction: If it just did a database and only detect the grammar in it, just need a few AI
  If it can detect the words meaning and using environment and give recommend use, it need a high extent AI
- Internet routing algorithms: I think just a few, but if it involves some dynamic encryption, it should be a lot of AI extent.

3. a) Can be solved, through the robotic arm and object detection. But, if the "decent" means play like Olympic Games, it can't be solved. The speed is too fast, and the core calculation can't reach that, maybe 10 years will get that fast

b) Can't be solved. Because the center of Cairo is too crowed. The auto driving now can't be so accurate to deal with such complex situation. It will solved also by increase the calculation speed, maybe several years as the hardware development

c) Can be

d) If it means auto drive a robot to market, buy, find, bargain, It can't be solved. It touchs too many areas and need a huge develop in both software and hardware

e) Can be

f) can be

g) can't be. The most difficult is the word discover, which means the machine at least need to continue update itself without human. It's hard.

h) can be

i) can be

j) can be

k) can't be. Maybe the machine can, but the human law didn't allow this now. Maybe 5~10 years will solve this problem. Also the improvement in accurate surgical control is necessary.

4. a) We use a triple (m, c, b) to represent the state on left river bank, m represent the number missionaries and c is the number of cannibals. $b=1$ means boat on left river bank, $b=0$ means not

limit set $m \geq c$ and $m+c \leq 2$ on boat $b \in \{0, 1\}$

The total number of state in state space is $(N+1) \times (N+1) \times 2$ and the intital state of problem is $(N, N, 1)$, our goal is to reach $(0, 0, 0)$

4. b) searching method: To avoid repetition, we need record the search state.

First, we must define the safe state:

safe state: 1) There are equal number of missionaries and cannibals

2) Missionaries are all on left bank

3) Missionaries are all not on left bank

Use the DFS search to the safe state ~~until~~ until find a legitimate solution

Also, we need set the heuristic function

$F_1(X) = M+C$

$F_2(X) = M+C-2B$

$F_1(X)$ satisfies the A algorithm, $F_2(X)$ satisfies the A^* algorithm

choose the large $F(X)$ as preferentially selected for search.

result show (one of example): $(3,3,1) \rightarrow (3,1,0) \rightarrow (3,2,1) \rightarrow (3,0,0)$

$\downarrow$

$(0,1,0) \leftarrow (0,3,1) \leftarrow (0,2,0) \leftarrow (2,2,1) \leftarrow (1,1,0) \leftarrow (3,1,1)$

$\downarrow$

$(0,2,1) \rightarrow (0,0,0)$

in all solutions: $F_1(X)$ real cost is 11, $F_2(X)$ real cost is 11

Actually in any situation, number of missionaries and cannibal, $F_2(X)$ will get the best solution.

c) I think the hard time is mainly on the bunch of choice in the search and the definition of the best search solution. The state space idea perfectly solve these problems.

5. a) False, sometimes the depth first search directly gives results or reach to the goal without backtracking, which will expand less nodes.

b) True, $h(n)=0$ will not show an exaggeration for the optimal distance towards the desire nodes

c) False, A^* is widely used in area unit for navigation

d) True, if there is a goal, it will happens at finite depth d and be found in $O(bd)$ steps

e) False. If a rook can pass multiple squares in one move, the Manhattan distance may overestimate the optimal remaining range of moves to reach the goal.

6. a) if all step cost are equal

$f(n) = g(n) = 1 \times$ Depth of n

$g(n)$ will be a multiple of depth n

and then BFS and uniform-cost search behave the same

is a special case

b) when $f(x) = 1/d(v)$, the best-first search acts the same with DFS

c) uniforme-cost search is A^* search with $h(n) = 0$

A^* : $f(n) = g(n) + \underset{0}{\underset{\uparrow}{h(n)}}$ $\Rightarrow$ $f(n) = g(n)$ uniform-cost search

7. $f(n) = (2-w)\, g(n) + w h(n)$

$= (2-w)\left[g(n) + \frac{w}{2-w} h(n)\right]$

$\frac{w}{2-w} \leq 1$

as $w \leq 1$ is the complete

and $0 \leq w \leq 1$ is optimal

For $w = 0$: $f(n) = 2g(n)$ uniform-cost search

$w = 1$: $f(n) = g(n) + h(n)$ A^* search

$w = 2$: $f(n) = 2h(n)$ Greedy best-first search

8. heuristic $h(n)$ = Manhattan Distance + value of the tile

$h^*(n)$ refer to the path has minimum cost path starting from n ending to the goal node.

$h = h_1 + h_2$, if $h(n) \leq h^*(n)$ and if G_2 is a goal that suboptimal More than c

$f_1(G_2) > c^* + c$

$f(n) = f_1(n) + h(n)$

which $h^*(n) \leq f_1(n) + h^*(n) + c \leq c^* + c \leq f_1(G_2)$

$\therefore$ no expansion of G_2 will have before the expansion of goal

9. Billions of state space.