Solutions to Lecture 10
Intelligent Systems Programming

Exercise 1

a)

Iter #	Current State	Obj. Value	Neighbors
1	<D,C,E,B,F>	27	• <C,D,E,B,F> obj. 25 • <E,C,D,B,F> obj. 28 • <B,C,E,D,F> obj. 23 * • <F,C,E,B,D> obj. 25
2	<B,C,E,D,F>	23	• <C,B,E,D,F> obj. 24 • <B,E,C,D,F> obj. 23 • <B,D,E,C,F> obj. 22 • <B,F,E,D,C> obj. 18 *
3	<B,F,E,D,C>	18	• <E,F,B,D,C> obj. 22 • <B,E,F,D,C> obj. 17 * • <B,F,D,E,C> obj. 22 • <B,F,C,D,E> obj. 24
4	<B,E,F,D,C>	17	• <F,E,B,D,C> obj. 20 • <B,F,E,D,C> obj. 18 • <B,E,D,F,C> obj. 20 • <B,E,C,D,F> obj. 23

b)

Iter #	Current State	Obj. Value	Neighbors
1	<D,C,E,B,F>	27	• <C,D,E,B,F> obj. 25 *
2	<C,D,E,B,F>	25	• <D,C,E,B,F> obj. 27 • <C,E,D,B,F> obj. 25 • <C,B,E,D,F> obj. 24 *
3	<C,B,E,D,F>	24	• <E,B,C,D,F> obj. 27 • <C,E,B,D,F> obj. 24 • <C,B,D,E,F> obj. 20 *
4	<C,B,D,E,F>	20	• <E,B,D,C,F> obj. 26 • <C,E,D,B,F> obj. 25 • <C,B,E,D,F> obj. 24 • <C,B,D,F,E> obj. 21

Exercise 2

a)

Current state	Best state	Action taken	Fringe	Tabu List
A	A	-	BC	<>
B	B	*a*	CFD	< *A* >
D	D	*b*	FJE	< A,B >
E	E	*c*	J	< A,B,D >
J	E	*f*	H	< A,B,D,E >
H	E	*g*	G	< A,B,D,E,J >
G	E	*h*	FI	< A,B,D,E,J,H >
F	E	*i*	-	< A,B,D,E,J,H,G >

b)

Current state	Best state	Action taken	Fringe	Tabu List
A	A	-	BC	<>
B	B	*a*	CFD	< *a* >
D	D	*b*	FJE	< a,b>
E	E	*c*	J	< a,b,c >
J	E	*f*	DHI	< a,b,c,f >
D	E	*d*	F	< a,b,c,f,d >
F	E	*o*	BG	< a,b,c,f,d,o >
B	E	*j*	C	< a,b,c,f,d,o,j >
C	E	*e*	AIK	< a,b,c,f,d,o,j,e >
K	K	*n*	-	< a,b,c,f,d,o,j,e,n >

Exercise 3

a)

In the following we consider the initial state s = (2, 1, 1, 2) of the 4-queen problem. The tree of possible states that the HILL-CLIMBING algorithm will explore from s is shown in Figure 1. The numbers on the game board show the value of the min-conflict heuristic for the queen in the specified column.

b)

Two of the leaves in the tree of Figure 1 are not goal nodes. These are labeled A and B respectively. For these nodes, the HILL-CLIMBING algorithm has reached a shoulder and is stuck; when the node A (B respectively) is chosen, it is chosen as a lowest-valued successor with the value 1. Since non of the successors of the state A (B) gives a value that is strictly less than 1, the algorithm stops. If we allow sideway moves, we can (in this case) ensure that the algorithm will reach a solution; this is seen in the exploration tree of figure 2.

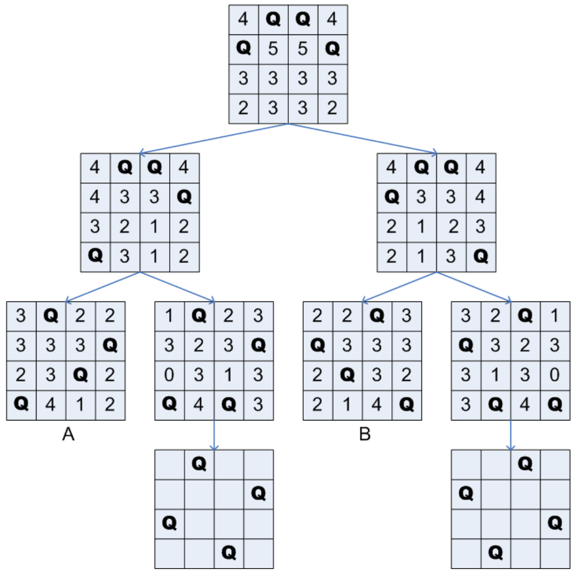

Figure 1

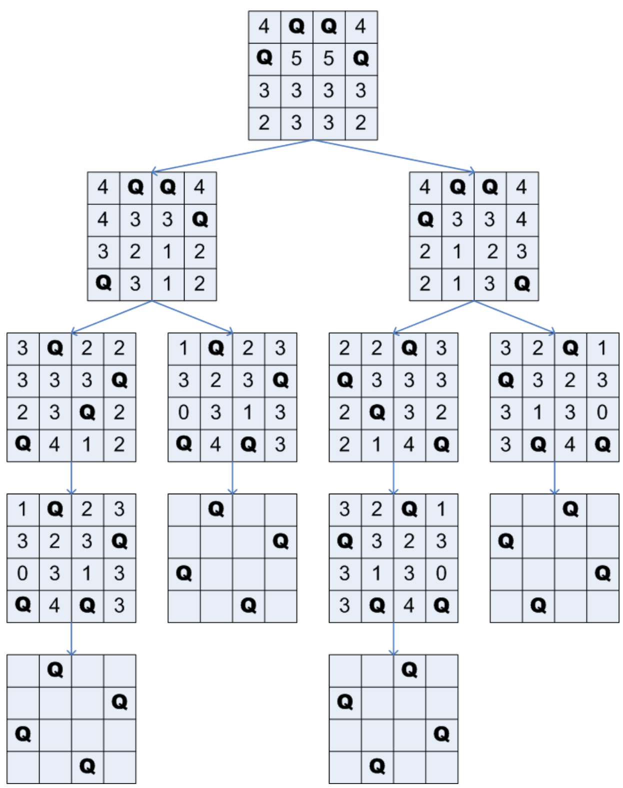

Figure 2