Class Exercise 2

Exercise 1 : Basic Concepts

(a) Let G be state-space graph (OR graph), and let n be a node in G. What is the difference between a solution **path** P for node n in G to a solution **base** P' for n in G?

(b) Which of the following statements is true?

- ☐ In an OR graph every solution path is a solution base.
- ☐ In an OR graph every solution base is a solution path.

Exercise 2 : Use of information in DFS

(a) Node expansion in step 5. of algorithm DFS is performed in a totally uninformed way. Which information can be used in node expansion and how can we use it?

(b) Does this also work for the backtracking algorithm bt?

(c) How can previously found solutions increase efficiency of DFS for optimization problems?

Exercise 3

(a) Describe the roles of the OPEN and CLOSED lists in breadth-first search.

(b) What is the role and functioning of the procedure clean-up-closed? Is the procedure necessary for depth-first search to work?

Exercise 4 : Size of OPEN and CLOSED lists

(a) Assume we are searching an uniform binary tree T using DFS with depth bound k and starting from the root node s. Consider the first loop cycle during the run where a node of depth l with $l < k$ will be processed. (The root node s has depth 0.)

How many nodes are on OPEN resp. on CLOSED at that point in time? (You may use the $\mathcal{O}$-notation to give your result.)

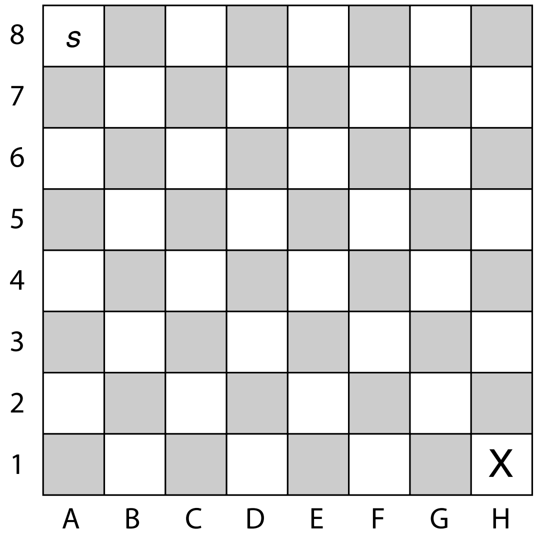
(b) Assume we are searching an uniform binary tree T using BFS and starting from the root node s. Consider the first loop cycle during the run where a node of depth l will be processed. (The root node s has depth 0.)

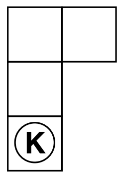
How many nodes are on OPEN resp. on CLOSED at that point in time? (You may use the $\mathcal{O}$-notation to give your result.)

(c) What are the consequences to your results, if we skip the precondition of T being uniform.

Exercise 5 : Knight Movess

Consider the Knight-Move Problem.





Knight move

The board consists of 64 squares named from A1 to H8. A knight moves two squares vertically and one square horizontally, or two squares horizontally and one square vertically.

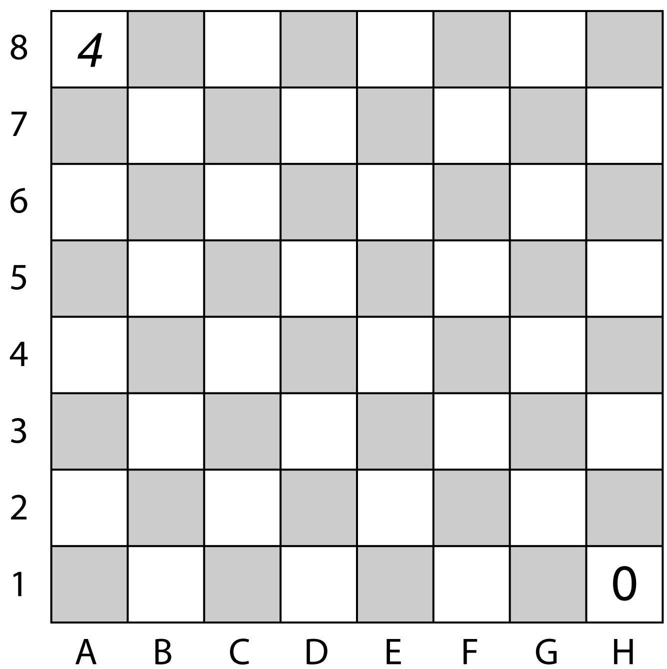
We are searching for a (shortest) sequence of knight moves leading from position $s = A8$ to position $X = H1$.

For that purpose we use the heuristic function

$$h = \max\left(\left\lceil \frac{\#\textit{rows}}{2} \right\rceil, \left\lceil \frac{\#\textit{columns}}{2} \right\rceil \right)$$

Here, $\#$*rows* denotes the row difference between current position of the knight and the target position, $\#$*columns* denotes the column difference between current position of the knight and the target position, e.g., $h(s) = \max(\lceil \frac{7}{2} \rceil, \lceil \frac{7}{2} \rceil) = 4$. ($\lceil . \rceil$ denotes the ceiling function.)

(a) Give the value of the heuristic function h for each cell on the board. (Note symmetry and regular formation.)



(b) Illustrate the processing of Basic-OR by filling out the following tables. The **first table** should show the content of OPEN and CLOSED at the beginning of the main loop (line 2. in Basic-OR pseudocode). Assume that Basic-OR selects a node with minimum h-value. **The node that will be selected next for expansion has to be underlined.** Nodes have to be described by $n_i(C5)$, where i is a unique index and the cell coordinates are given in brackets.

In the **second table** the h-values of **ALL successor nodes** of the expanded node should be listed. (The same cell can occur multiple times in the second table!) Separate the node expansions by **horizontal lines**.

Perform 3 node expansions. (The last line in the first table should read 3. .....)

Loop	OPEN	CLOSED
0.	$\underline{s = n_0(A8)}$	—
1.		

n	$h(n)$
$n_0(A8)$	4