情報工学演習Ｉ　「認知システム論」　2015 年 5 月 25 日

- 「探索」に関する以下の４問に解答し，授業時間内に提出してください.
- 時間内に提出できないときは，5 月 29 日（金）までに佐藤助教（8-03 室）に提出してください.

$\boxed{1}$　以下の文の空所にあてはまる適切な用語を答えなさい.

　探索問題は，１つの$\boxed{\quad 1 \quad}$，$\boxed{\quad 2 \quad}$の集合，$\boxed{\quad 3 \quad}$検査手続き，および$\boxed{\quad 4 \quad}$の４種類の情報からなる．$\boxed{\quad 1 \quad}$から$\boxed{\quad 3 \quad}$に達するまでの$\boxed{\quad 2 \quad}$の列を，その探索問題の解という.

　一般的な探索アルゴリズムは，$\boxed{\quad 5 \quad}$と呼ばれるデータ構造を用いて，探索を制御する．まず，アルゴリズムの実行開始時点で，$\boxed{\quad 1 \quad}$を表すノードを生成し，探索木の$\boxed{\quad 6 \quad}$とする．つぎに，その$\boxed{\quad 6 \quad}$に対して適用可能な$\boxed{\quad 2 \quad}$をすべて適用してすべての後続状態を生成し，その１つ１つを表すノードを生成する．そして，$\boxed{\quad 6 \quad}$を$\boxed{\quad 7 \quad}$，生成された各ノードを$\boxed{\quad 8 \quad}$とし，その２つのノードを，親子関係を表す辺で結ぶ.

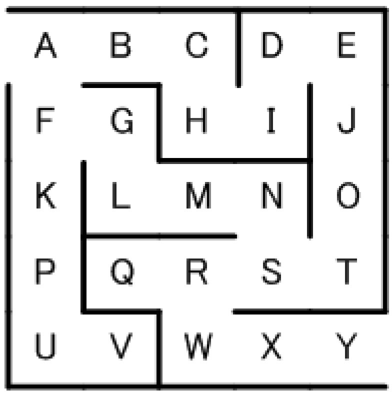
　今度は，いま生成された$\boxed{\quad 8 \quad}$のそれぞれが$\boxed{\quad 6 \quad}$の役割を果たして同様な処理を続ける．すなわち，一般に，アルゴリズムは探索木の$\boxed{\quad 9 \quad}$のどれか１つを$\boxed{\quad 7 \quad}$として選び，そのノードが表す状態に対して適用可能なオペレータをすべて適用してすべての後続状態を生成し，その１つ１つを表す$\boxed{\quad 8 \quad}$を生成し，親と子を辺で結ぶことによって，探索木を成長させていく．$\boxed{\quad 7 \quad}$から$\boxed{\quad 8 \quad}$の集合を作る操作を$\boxed{\quad 10 \quad}$という．アルゴリズムの基本動作は，このようにノードを次々と$\boxed{\quad 10 \quad}$して$\boxed{\quad 5 \quad}$を成長させていき，$\boxed{\quad 3 \quad}$ノードが生成されるのを待つことである.

　このような探索アルゴリズムの実装は，ノードのリストを用いたつぎのような考え方で行うことができる．まず，$\boxed{\quad 5 \quad}$に含まれるノードは未$\boxed{\quad 10 \quad}$か$\boxed{\quad 10 \quad}$済みかのいずれかである．これらのノードを分けて，それぞれリストに並べて保持しておく．未$\boxed{\quad 10 \quad}$ノードのリストを$\boxed{\quad 11 \quad}$リストと呼び，$\boxed{\quad 10 \quad}$済みノードのリストを$\boxed{\quad 12 \quad}$リストと呼ぶ．$\boxed{\quad 11 \quad}$リストの中から$\boxed{\quad 7 \quad}$を選ぶときには先頭から選び，それを$\boxed{\quad 11 \quad}$リストから取り除いて$\boxed{\quad 12 \quad}$リストに移す.

　一方，生まれた$\boxed{\quad 8 \quad}$を$\boxed{\quad 11 \quad}$リストに追加するときには，何らかの$\boxed{\quad 13 \quad}$に基づいて適切な位置に挿入する．その特別な場合として，$\boxed{\quad 8 \quad}$を常に先頭に追加する場合は，このリストは，$\boxed{\quad 14 \quad}$と呼ばれる$\boxed{\quad 15 \quad}$型(LIFO)のデータ構造となり，そのアルゴリズムは$\boxed{\quad 16 \quad}$探索となる．一方，$\boxed{\quad 8 \quad}$を常に末尾に追加する場合は，リストは$\boxed{\quad 17 \quad}$と呼ばれる$\boxed{\quad 18 \quad}$型(FIFO)のデータ構造となり，そのアルゴリズムは$\boxed{\quad 19 \quad}$探索となる.

　ノードはいくつかのフィールドからなる構造データとして実装し，そのうちの１つのフィールドには，$\boxed{\quad 7 \quad}$を指し示す$\boxed{\quad 20 \quad}$を格納しておけば，$\boxed{\quad 3 \quad}$ノードを発見したときに，それをたどれば$\boxed{\quad 6 \quad}$までの経路が得られるので，それを逆順に出力したものが，探索問題の解となる.

$\boxed{2}$ 下図の迷路に関して，A から Y までの 25 個の状態からなる状態空間を考え，入口 A から出口 Y までの経路を発見する探索問題を考える．迷路の中では，A→G のように斜めには進めないものとする．また，A→B→A のように直前の地点にただちに戻ることはしないものとする．



以下の設問に答えなさい．ただし，設問(1)～(3)においては，探索方向に迷ったとき（親ノードの評価値が同じであるため，選択を一意に行えないとき）は，アルファベット順が小さいもの（たとえば，F より B）を優先するものとする．また，異なる２つのノードが同じ状態（地点）を表すものかどうかはチェックせず，そのようなノードは探索木に重複して出現するものとする．

(1) 幅優先探索によって経路を求めたときの探索木を示し，探索の順番（展開したノードの順番）をそのノードに付記しなさい．

(2) 深さ優先探索では経路が求められないことを説明しなさい．また，深さ優先探索によって経路を求めるためのアルゴリズムの修正案を簡単に述べ，実際に経路が求められたときの探索木を示し，探索の順番をそのノードに付記しなさい．

(3) A*アルゴリズムによって経路を求めたときの探索木を示し，探索の順番をそのノードに付記しなさい．ただし，評価関数は，$f(n) = g(n) + h(n)$ とする．$g(n)$ は A から地点 n までの移動距離である．$h(n)$ はヒューリスティック関数であり，迷路に壁がないと仮定したときの地点 n から Y までの移動距離（マンハッタン距離）である．

$\boxed{3}$ つぎのように定式化された探索問題について考える.

- 状態の集合は，自然数全体の集合 $\{1,2,3,\cdots\}$ である.
- 初期状態は，1 である.
- オペレータとして，以下に示す A,B の 2 つがある.

$$A(x)=x+1,\quad B(x)=3x$$

すなわち，A は現在の状態に 1 を加えた状態に遷移させ，B は現在の状態を 3 倍した状態に遷移させる. たとえば, 初期状態 1 から最初に A, 次に B を適用すると, 状態は $1\to2\to6$ と遷移する.

- 各状態遷移には 1 だけコストがかかり，状態遷移列（経路）のコストはその状態遷移の回数（経路の長さ）に等しい. たとえば，上記の経路 $1\to2\to6$ のコストは 2 である.
- 目標状態（ゴール）は，状態 12 である.

(1) この探索問題を A* アルゴリズムで解いたときに，最初に見つかる解を示しなさい. 解が得られた時点での探索木も，あわせて図示すること. ただし，状態 n から目標状態までの最小コストの見積もり，すなわちヒューリスティック関数をつぎのように定義する.

$$h(n)=\begin{cases}12-n & \text{if } n\le 12\\ \infty & \text{if } n>12\end{cases}$$

(2) 上で求めた解は最適解（最小コストの解）ではないことを示しなさい.

$\boxed{4}$　つぎのような初期配置のブロック並べパズルについて考える．

B	B	W	W	

黒い駒$\boxed{\mathrm{B}}$が２個，白い駒$\boxed{\mathrm{W}}$が２個と空所$\boxed{}$が１カ所ある．このパズルは，つぎのようにプレイする．

(a) １個の駒は，隣の空所へ単位コスト（1.0）で移動できる．

(b) １個の駒は，ちょうど２個の駒を飛び越して空所へ移動できる．そのときのコストは，1.1 である．（駒を１個だけや３個以上飛び越すことはできない．）

したがって，配置の変化を起こすための可能なオペレータはつぎの４つである．

L：空所の左隣の駒（もしあれば）を空所にコスト 1.0 で移動する．

R：空所の右隣の駒（もしあれば）を空所にコスト 1.0 で移動する．

3L：空所の３つ左にある駒（もしあれば）を空所にコスト 1.1 で移動する．

3R：空所の３つ右にある駒（もしあれば）を空所にコスト 1.1 で移動する．

たとえば，初期配置でオペレータ 3L を適用すると，つぎのようになる．

B		W	W	B

パズルの目標は，すべての白い駒$\boxed{\mathrm{W}}$がすべての黒い駒$\boxed{\mathrm{B}}$の左に置かれることである．（空所は無視する．）

このパズルを探索問題として扱い，探索アルゴリズムで解いてみよう．

(1) この問題について，深さ４（すなわち，オペレータの適用回数が４回）の探索木を図示しなさい．ただし，直前の状態にただちに戻るような状態遷移は考えない．たとえば，L を適用した直後に R を適用したり，3L を適用した直後に 3R を適用したりしない．

ここで作成した探索木の各ノード n に初期状態から n までの経路コスト $g(n)$を付記しなさい．また，ノード n から目標状態までの最短コストの見積もりを表すヒューリスティック関数 $h(n)$の案を１つ考え，$h(n)$の値，および総経路コストの見積もり $f(n)=g(n)+h(n)$の値を各ノードに付記しなさい．$h(n)$の案が１つも思い付かないときは，探索木のすべてのノード n について $h(n)=0$ としなさい．ただし，その場合には，設問(3)(4)において，かなり大きな探索木になることがあるので覚悟が必要である．

(2) 均一コスト探索（$g(n)$を評価値とする最良優先探索）でこの問題を解き，解が得られたときの探索木を示しなさい．

(3) 欲張り最良優先探索（$h(n)$を評価値とする最良優先探索）でこの問題を解き，解が得られたときの探索木を示しなさい．

(4) A^*アルゴリズム（$f(n)$を評価値とする最良優先探索）でこの問題を解き，解が得られたときの探索木を示しなさい．

$\boxed{\text{ヒント}}$　探索木の深さ４の位置に目標状態（ゴール）の１つがあり，ここに至る解が最適解（コスト 4.2）である．(1)で得られる探索木のノード数は 19 である．(2)(3)(4)では，最良優先探索において評価値が最小である複数のノードからどれを選んで展開するかの決定や，ヒューリスティック関数の設定に依存して結果は異なるが，出題者がこの問題を解いたときには，得られた探索木のノード数は，(2)では 25，(3)では 11，(4)では 15 であった．(2)と(4)では最適解が得られ，(3)では深さ 7 の位置にある非最適解（コスト 7.3）が得られた．