データ構造とプログラミング技法　　学科＿＿＿＿学籍番号＿＿＿＿＿＿氏名＿＿＿＿＿＿＿＿＿＿＿＿

1. 以下の文章の空欄を埋めなさい．(1 問 1 点：合計 24 点)

- 関係 R について反射律（ xRx ），［　　］律（ xRyかつyRzならばxRz ），反対称律（［　　］）の 3 つの関係が常に成り立つとき関係 R を半順序関係と呼ぶ．これらの条件に加えて（［　　］または［　　］）が常に成り立つとき，関係 R を全順序関係と呼ぶ．
- 全順序関係は［　　］構造，半順序関係は木構造や，［　　］構造における要素間の関係である．また，一般的な 2 項関係は［　　］構造という論理構造によって表現することができる．
- 物理構造としては，一連のデータを連続するメモリ番地に記憶する［　　］配置，メモリ上に分散したデータをポインタによって接続する［　　］配置の 2 種類がある．
- ヒープソートでは，キー列のデータ構造を，論理構造としては［　　］，物理構造としては配置と見なしている．
- アクセスの方法が事前に決められた線形構造の例としては［　　］，［　　］，デックがある．これらのデータ構造に対する操作は，データの追加と削除が許されている．このようにデータ構造とそれに対する操作手続きを組にして定義することを［　　］化と呼ぶ．
- 再起呼び出しを用いた木構造の縦型探索アルゴリズムを，繰り返し計算を用いたアルゴリズムに変換する際には，［　　］という線形構造を用いる．また，木構造の横型探索アルゴリズムを実現するには［　　］を用いる．
- ハッシュ法における衝突処理は開番地法と［　　法］に分類されるが，開番地法の一つである［　　走査法］では，表サイズとハッシュ増分が互いに素である場合，探索周期は［　　］になる．
- k 次の B 木は，根には［　　］個以上［　　］個以下のキーが格納され,それ以外の節には［　　］個以上［　　］個以下のキーが格納され,全ての葉は［　　］にあるというデータ構造である．

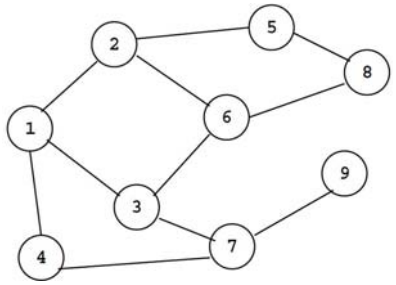
2. 下記はグラフ探索を行うプログラムである．空白部分を埋めなさい．(1 問 2 点：合計 6 点)

```
graph-search(vertex v)
{    push(S,v);
     while (!empty(S)) { [　　];
                         if ([　　==　　]) { v.visit := 'YES' ;
                              for (i=0; i<N;i++)
                                   if ((adj[w[i]][v]==1)&&([　　==　　])) push(S,w[i]) ;
                         }
     }
}
```

但し，

- N は頂点数．w[i]は i 番目の頂点．adj[w][v]は隣接行列であり，頂点 w,v 間にエッジがあれば 1，なければ 0 の値がセットされている．
- 線形構造 S に対しては，vertex 型のデータ v を格納する push(S,v)と,v を取り出す pop(S,v)の操作のみが許されている．
- vertex 型のデータ v の visit 欄の初期値は'NO'であるとする．

3. 上記の関数で, 下図のグラフの⑤を起点として探索を行った場合, 節の番号を訪れる順番に並べなさい. 但し, グラフの各節に書いてある番号 i はその頂点が i 番目であることを表している．(6 点)



4. グラフの中心を求めたい．使用するアルゴリズムと，その結果の利用法について説明しなさい．（4 点）

5. 年功序列を基本とする給与体系を採用している会社の社員について，氏名，年齢，給与の 3 つの欄から成る名簿データがある．ソーティングを 2 回行い，このデータを年齢の小さい順に並べ，且つ，同じ年齢の場合は給与が少ない順に並ぶようにしたい．1 回目と 2 回目のソーティングに用いる欄について説明し，2 回目のソーティングで使用すべきアルゴリズムとして何が適当かを理由も含めて説明しなさい．（5 点）

6. 以下に示すアルゴリズムの名称を述べなさい．また空白を埋めなさい．（空白 1 点，名前 2 点：合計 3 点）

```
last = 1;
while (sw!=n-1) { sw = n-1;
                  for (i=n-1; i>last; i--)
                   if (a[i]<a[i-1]) {
                            w=a[i]; a[i]=a[i-1];a[i-1]=w;
                   }
                  last = [          ];
}
```

7. キーが 5, 15, 19, 1, 6, 11, 10, 9, 14 の順に配列 A[1]～A[9]に格納されているものとする．この配列が順配置された完全 2 分木である場合，その木を図示しなさい．また，これをヒープに変更した場合の木構造を示しなさい．（4 点＋6 点：合計 10 点）

8. 2, 5,10,3,15, 19,14,11,8,1 を順に与えたときのハッシュ表と成功探索時の平均探査回数を求めなさい．但し，表サイズは 11，$h_0(x) = x \bmod 11, h_i(x) = (h_0(x) + 2 \times i) \bmod 11$ とする．（表 5 点，平均探査回数 3 点：合計 8 点）

表										
探査回数										

平均探査回数=　　回

9. 21, 2, 35, 14, 17, 5, 6, 20 というデータを順に与えたときに出来上がる二分探索木と，平衡が崩れた際に再並行化を行って作成した AVL 木を図示し，成功探索時の平均探査回数をそれぞれ求めなさい．(2 分探索木 2 点，AVL 木 4 点，平均探査回数は各 2 点：合計 10 点）

平均探査回数=　　回

平均探査回数=　　回

10. テキスト ABCABABBBABABBA から文字列 ABABBA を見つけるための文字比較回数は，単純照合法，KMP 法を用いた場合でそれぞれどのようになるか答えなさい．（2 点+6 点：合計 8 点）

ABCABABBBABABBA

単純照合法：　　回

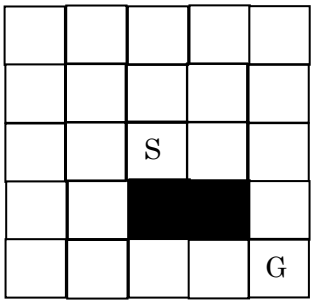
ABCABABBBABABBA

KMP 法　　回

11. 図の $\boldsymbol{S}=(x_S, y_S)$ からスタートして $\boldsymbol{G}=(x_G, y_G)$ に至る最短経路探索問題を考える．各マスにおいて移動可能な方向は，上下左右と斜め方向で，1 回の移動コストは，上下左右が 1，斜め方向が $\sqrt{2}$ である．探索の過程で，$\boldsymbol{S}$ から $\boldsymbol{M}=(x_M, y_M)$ に到達している場合，そこから $\boldsymbol{G}$ に到る緩和解の評価関数を

$$g(\boldsymbol{M}) = Cost(\boldsymbol{S}, \boldsymbol{M}) + \max(|x_M - x_G|, |y_M - y_G|) + (\sqrt{2} - 1)\min(|x_M - x_G|, |y_M - y_G|)$$

とする．但し，$Cost(\boldsymbol{S}, \boldsymbol{M})$ は $\boldsymbol{S} \Leftrightarrow \boldsymbol{M}$ の間の実移動コストの累積値である．このとき，best-first の縦型探索を用いて，分枝限定法で最短経路の探索を行う場合，探索が行われないマス（そのマスまでの実移動コストが計算されないマス）を塗りつぶし，最短経路の総移動コストを求めなさい．但し，黒いマスは障害物を表わす．(16 点)



データ構造とプログラミング技法　　学科＿＿＿＿学籍番号＿＿＿＿＿＿氏名＿＿＿＿＿＿＿＿＿＿＿＿

1. 以下の文章の空欄を埋めなさい．（1 問 1 点：24 点）

- 関係 R について反射律（ xRx ），［推移律］（ xRyかつyRzならばxRz ），反対称律（［xRyかつyRxならば$x=y$］）の 3 つの関係が常に成り立つとき関係 R を半順序関係と呼ぶ．これらの条件に加えて（［xRy］または［yRx］）が常に成り立つとき，関係 R を全順序関係と呼ぶ．
- 全順序関係は［線形］構造，半順序関係は木構造や，［束］構造における要素間の関係である．また，一般的な 2 項関係は［グラフ］構造という論理構造によって表現することができる．
- 物理構造としては，一連のデータを連続するメモリ番地に記憶する［順］配置，メモリ上に分散したデータをポインタによって接続する［リンク］配置の 2 種類がある．
- ヒープソートでは，キー列のデータ構造を，論理構造としては［完全二分木］，物理構造としては［順］配置と見なしている．
- アクセスの方法が事前に決められた線形構造の例としては［スタック］，［キュー］，デックがある．これらのデータ構造に対する操作は，データの追加と削除が許されている．このようにデータ構造とそれに対する操作手続きを組にして定義することを［データ抽象］化と呼ぶ．
- 再起呼び出しを用いた木構造の縦型探索アルゴリズムを，繰り返し計算を用いたアルゴリズムに変換する際には，［スタック］という線形構造を用いる．また，木構造の横型探索アルゴリズムを実現するには［キュー］を用いる．
- 衝突処理は開番地法と［連鎖　法］に分類されるが，開番地法の一つである［線形 走査法］では，表サイズとハッシュ増分が互いに素である場合，探索周期は［全表的］になる．
- k 次の B 木は，根には［1］個以上［2k］個以下のキーが格納され,それ以外の節には［k］個以上［2k］個以下のキーが格納され,全ての葉は［同一レベル］にあるというデータ構造である．

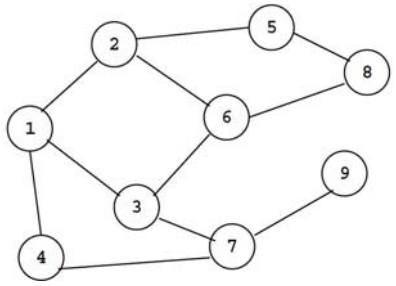
2. 下記はグラフ探索を行うプログラムである．空白部分を埋めなさい．（1 問 2 点：合計 6 点）

```
graph-search(vertex v)
{    push(S,v);
     while (!empty(S)) { [ pop(S,v) ];
                         if ( [ v.visit == 'NO' ] ) {   v.visit := 'YES';
                              for (i=0; i<N;i++)
                                   if ((adj[w[i]][v]==1)&&( [ w[i].visit== 'NO' ])) push(S,w[i]);
                         }
     }
}
```

但し，

- N は頂点数．w[i]は i 番目の頂点．adj[w][v]は隣接行列であり，頂点 w,v 間にエッジがあれば 1，なければ 0 の値がセットされている．
- 線形構造 S に対しては，vertex 型のデータ v を格納する push(S,v)と,v を取り出す pop(S,v)の操作のみが許されている．
- vertex 型のデータ v の visit 欄の初期値は'NO'であるとする．

3. 上記の関数で, 下図のグラフの⑤を起点として探索を行った場合, 節の番号を訪れる順番に並べなさい. 但し, グラフの節に書いてある番号 i はその頂点が i 番目であることを表している．（6 点）



5→8→6→3→7→9→4→1→2

4. グラフの中心を求めたい．使用するアルゴリズムと，その結果の利用法について説明しなさい．（4 点）

Floid のアルゴリズムを用いて，各頂点間の最短距離を求める．頂点 i から頂点 j への最短距離が，D[i,j]という形式で得られたとすると，頂点 i について $d[i]=\max_j D[i,j]$とする．全頂点の中で，d[i]が最も小さくなる頂点集合を求めると，それがグラフの中心になっている．

5. 年功序列を基本とする給与体系を採用している会社の社員について，氏名，年齢，給与の 3 つの欄から成る名簿データがある．ソーティングを 2 回行い，このデータを年齢の小さい順に並べ，且つ，同じ年齢の場合は給与が少ない順に並ぶようにしたい．1 回目と 2 回目のソーティングに用いる欄について説明し，2 回目のソーティングで使用すべきアルゴリズムとして何が適当かを理由も含めて説明しなさい．（5 点）

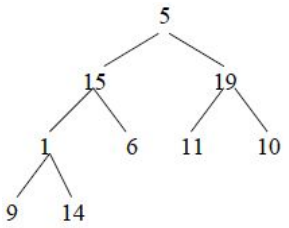
まず 1 回目は，給与についてソーティングを行い，2 回目は年齢についてソーティングを行う.この場合,2 回目のソートは安定なソートでなければならない．安定なソートには単純挿入法，単純交換法，基数ソートなどがあるが，この場合には年功序列であるので，1 回目のソートで概ソート列が得られているため，単純挿入法などが適している．

6. 以下に示すアルゴリズムの名称を述べなさい．また空白を埋めなさい．（空白 1 点，名前 2 点：合計 3 点）

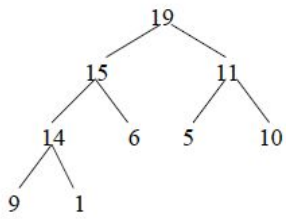
```
last = 1;
while (sw!=n-1) { sw = n-1;
                  for (i=n-1; i>last; i--)
                     if (a[i]<a[i-1]) {
                                 w=a[i]; a[i]=a[i-1];a[i-1]=w;
                     }
                  last = sw+1 ;
}
```

これはバブルソートアルゴリズムである．

7. キーが 5, 15, 19, 1, 6, 11, 10, 9, 14 の順に配列 A[1]～A[9]に格納されているものとする．この配列が順配置された完全 2 分木である場合，その木を図示しなさい．また，これをヒープに変更した場合の木構造を示しなさい．（4 点＋6 点：合計 10 点）



完全二分木



ヒープ

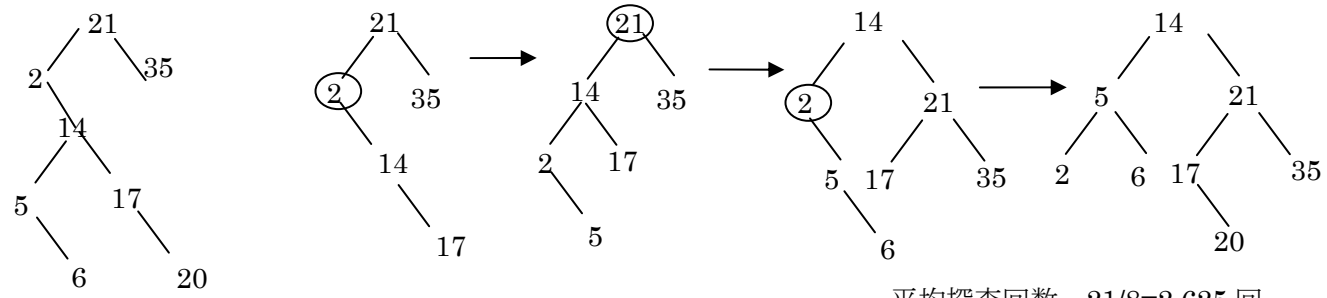
8. 2, 5,10,3,15, 19,14,11,8,1 を順に与えたときのハッシュ表と成功探索時の平均探査回数を求めなさい．但し，表サイズは 11，$h_0(x) = x \bmod 11, h_i(x) = (h_0(x) + 2 \times i) \bmod 11$ とする．（表 5 点，平均探査回数 3 点：合計 8 点）

表	11	8	2	3	15	5		14	19	1	10
探査回数	1	3	1	1	1	1		3	1	5	1

合計 18 回

平均探査回数=18/10=1.8 回

9. 21, 2, 35, 14, 17, 5, 6, 20 というデータを順に与えたときに出来上がる二分探索木と，平衡が崩れた際に再並行化を行って作成した AVL 木を図示し，成功探索時の平均探査回数をそれぞれ求めなさい．(2 分探索木 2 点，AVL 木 4 点，平均探査回数は各 2 点：合計 10 点)



平均探査回数　26/8=3.25 回

平均探査回数　21/8=2.625 回

10. テキスト ABCABABBBABABBA から文字列 ABABBA を見つけるための文字比較回数は，単純照合法，KMP 法を用いた場合でそれぞれどのようになるか答えなさい．（2 点+6 点：合計 8 点）

```
ABCABABBBABABBA
ABA
 A
  A
   ABABBA
    A
     ABA
      A
       A
        A
         ABABBA
```

単純照合法：24 回

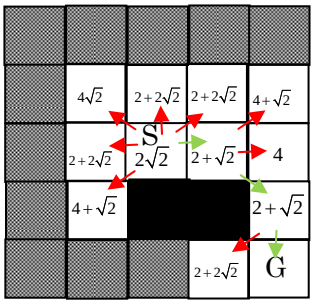
```
ABCABABBBABABBA
ABA
    ABABBA
          ABABBA
```

KMP 法 15 回

11. 図の $\boldsymbol{S}=(x_S, y_S)$ からスタートして $\boldsymbol{G}=(x_G, y_G)$ に至る最短経路探索問題を考える．各マスにおいて移動可能な方向は，上下左右と斜め方向で，1 回の移動コストは，上下左右が 1，斜め方向が $\sqrt{2}$ である．探索の過程で，$\boldsymbol{S}$ から $\boldsymbol{M}=(x_M, y_M)$ に到達している場合，そこから $\boldsymbol{G}$ に到る緩和解の評価関数を

$$g(\boldsymbol{M})=Cost(\boldsymbol{S},\boldsymbol{M})+\max(|x_M-x_G|,|y_M-y_G|)+(\sqrt{2}-1)\min(|x_M-x_G|,|y_M-y_G|)$$

とする．但し，$Cost(\boldsymbol{S},\boldsymbol{M})$ は $\boldsymbol{S}\Leftrightarrow\boldsymbol{M}$ の間の実移動コストの累積値である．このとき，best-first の縦型探索を用いて，分枝限定法で最短経路の探索を行う場合，探索が行われないマス（そのマスまでの実移動コストが計算されないマス）を塗りつぶし，最短経路の総移動コストを求めなさい．但し，黒いマスは障害物を表わす．(16 点)



右図のように探索が行われる．（最初は S から見て 6 方向に進み，best first サーチなので $g(\boldsymbol{M})=2+\sqrt{2}$ となる右向き緑色の矢印の方向が選ばれる．右に移動した後，未訪問の 3 方向（右，右上，右下）のコストが計算される．このうち $g(\boldsymbol{M})=2+\sqrt{2}$ となる右下向き緑の矢印方向が選ばれる．その方向に移動し，2 方向（下，左下）が調べられ，下方向が G なので，この時点で暫定コスト $z=2+\sqrt{2}$ が求められる．既に展開された各ノードの $g(\boldsymbol{M})$ は全て z の値よりも大きいので，全て終端されてしまい，これ以上探索は行われない．したがって，灰色のマスが探索されない．