アルゴリズム論・試験問題

問題 1 以下は Knuth-Moriss-Pratt のパターン照合アルゴリズムでパターン照合のための DFA を構成するための擬似コードである．

入力: $\Sigma = \{b_1, ..., b_l\}$ 上のパターン $p = a_1 \cdots a_m$

出力: パターン照合のための DFA $A = (Q, \Sigma, \delta, p_0, F)$

ConstDFA(p,m)
{
  $Q := \{0, 1, ..., m\}$ とする;
  $F := \{m\}$ とする; $p_0 := 0$ とする;
  for$(i := 1; i \le m; i++)$ $\delta(i-1, \boxed{\text{(a)}}) := i$ と設定;
  for$(i := 1; i \le l; i++)$ **if**$(b_i \ne a_1)$ $\delta(0, b_i) := \boxed{\text{(b)}}$ と設定;
  $f :=$**compF**(p,m);
  /* 決定性有限オートマトン (DFA) を作る */
  for$(i := 1; i \le m; i++)$
    for$(j := 1; j \le l; j++)$
      if$(b_j \ne a_{i+1})$ $\delta(i, b_j) := \delta(\boxed{\text{(c)}}, b_j)$ と設定;
}

ここで，**compF** は失敗処理関数 f を計算する関数である．

(1) 擬似コードの空欄 (a), (b), (c) を埋めよ．

(2) 失敗処理関数 f の定義を述べなさい．ただし，$\prec_s$ という記号を用いてよい．ここで，$x \prec_s y$ は x が y の真の接尾語であることを表す．

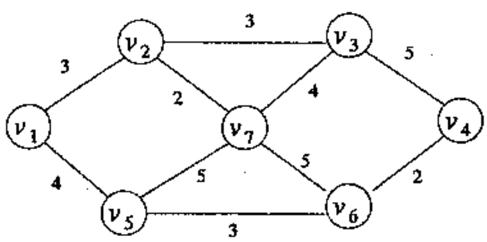
(3) パターン $p = aabaab$ のパターン照合を行うことを考える．このパターンに対する失敗処理関数の値を求めよ．

(4) (3) で与えたパターンに対するパターン照合のための DFA を書け．ただし，$\Sigma = \{a, b\}$ とする．

(5) 文字列 $w_1, w_2 \in \Sigma^*$ を考える．ある文字列 $\alpha, \beta \in \Sigma^*$ に対して，$w_1 = \alpha\beta$, $w_2 = \beta\alpha$ と書けるとき，w_2 は w_1 の回転であるという．たとえば，abc と cab は互いに他方の回転である．2 つの文字列 p_1 と p_2 が与えられたとき，p_2 が p_1 の回転になっているかどうかを決定するための線形時間のアルゴリズムを考えよ．

問題 2 次の Prim のアルゴリズムと重み付き無向グラフ G に対して以下の設問に答えよ．ただし，V を G の頂点集合 $\{v_1, v_2, v_3, v_4, v_5, v_6, v_7\}$ とする．また，辺は，$\{v_1, v_2\}$ のように，辺の端点 2 つの要素からなる集合で表すことにする．

```
任意に選んだ頂点 r のキーの値を 0,
それ以外の頂点のキーの値を∞として,
プライオリティキュー Q を構成する.
parent[r] = NIL;
while(Q!=φ){
  u = EXTRACT-MIN(Q);
  for each v in Adj[u] do
    if( v in Q かつ w(u,v) < key[v]){
      parent[v] = u;
      key[v] = w(u,v);
    }
  end
}
```



重み付き無向グラフ G

ここで，w(u,v) は辺 $\{u, v\}$ の重みを表し，EXTRACT-MIN は，プライオリティキューから最小のキーを持つ要素を抽出する操作であり，Adj[u] は頂点 u が隣接する頂点の集合を表す．

(1) $V' = \{v_1, v_2, v_7\}$ とし，カット $C = (V', V - V')$ を考える．C と交差する辺をすべて枚挙せよ．

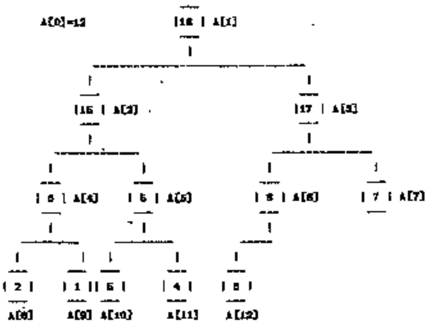
(2) 上記のカット C に対する軽い辺を求めよ．

(3) プライオリティーキューとはどのようなデータ構造かを説明せよ.

(4) グラフ G に対して Prim のアルゴリズムを適用することを考える. このとき, 以下の空欄を埋めよ.

1 行目で r として v_1 が選ばれたとする. このとき, 最初に 4 行目で u として選ばれるのは, 頂点 $\boxed{\text{(a)}}$ である. そして, 5〜9 行目の設定の結果, key[v_2] = $\boxed{\text{(b)}}$, key[v_5] = $\boxed{\text{(c)}}$ となる. よって, 次に 4 行目で u として選ばれるのは, 頂点 $\boxed{\text{(d)}}$ である. この u に対して, 5〜9 行目の設定の結果, key[v_3] = $\boxed{\text{(e)}}$, key[v_7] = $\boxed{\text{(f)}}$ となる. その後, 4 行目で u として選択される頂点は, $\boxed{\text{(g)}}$, $\boxed{\text{(h)}}$, $\boxed{\text{(i)}}$, $\boxed{\text{(j)}}$, $\boxed{\text{(k)}}$ という順番になる.

問題 3 配列を用いて 2 分ヒープを表現することを考える. ただし, ヒープ条件は,「親の値が子の値以上」とし, 配列 A の 添字 1 から 順番に要素を格納し, A[0] にはヒープを構成する頂点数を格納するものとする. 以下は, その例である. また, 配列要素 A[i] に対応する頂点を頂点 A[i] と呼ぶことにする.



頂点 A[i] の左側の子以下の部分木と右側の子以下の部分木がヒープの条件を満たしているとき, 頂点 A[i] 以下の部分木をヒープ化するプログラム heapify(int *A, int i) を考える.

```
void heapify(int *A, int i)
{
  int l,r,largest,x;

  l = left(i);
  r = right(i);

  if(l<=A[0] && A[l]>A[i]) largest = l;
  else largest = i;
  if(r<=A[0] && A[r]>A[largest]) largest = r;

  if(largest!=i){
    x = A[i];
    A[i] = A[largest];
    A[largest] = x;
    heapify(A,largest);
  }
}
```

(1) ヒープを構成する配列 A の値が, 以下のように初期化されているとき, 頂点 A[3] の左の子と右の子の値をそれぞれ求めなさい.

```
int a[] = {10,20,18,19,10,7,9,6,8,7,3};
```

(2) 関数 int left(int i) と int right(int i) のプログラムを書きなさい. ただし, left は頂点 A[i] の左の子の添字を求める関数であり, right は頂点 A[i] の右の子の添字を求める関数である.

(3) ヒープを構成する配列 A の値が, 以下のように初期化されているとき, heapify(A,1) を実行した後の配列 A の各要素の値 (A[0] 〜 A[12]) を示しなさい.

```
int a[] = {12,9,15,14,9,12,9,6,8,7,9,5,8};
```

(4) 与えられた配列 int *A をヒープ化する関数 void buildheap(int *a) を heapify を利用して書きなさい.

(5) ヒープ条件を「親の値が子の値以下」と変更したとき, プログラム heapify はどの箇所をどのように変更すべきか.