アルゴリズムとデータ構造　2018 年度冬学期　期末試験

※　計算や解答の下書きなどは計算用紙で行い、解答用紙には解答をよく整理して読みやすく記載せよ。

問1　ヒープ、2 分探索木、２－３木、ハッシュについて以下の問いに答えよ。答えは、要素数 n の式として示せ。答えのみでよい。(各 4 点,　計 12 点)

1)　新しい要素を挿入する際の最悪計算量を答えよ。

2)　最小の要素を取り出すときの最悪計算量を答えよ。

3)　特定の要素を削除する際の最悪計算量を答えよ。

問2　すでに n 個の要素が格納されている配列の要素を入れ替えてヒープ（半順序のついた 2 分木）にしたい。なるべく効率のよい方法を使うものとして、以下の問いに答えよ。(各 4 点,　計 12 点)

1)　アルゴリズム（手順）を簡潔に説明せよ。

2)　上記のアルゴリズムの計算量を答えよ。答えのみでよい。

3)　上記の計算量の導出の過程を示せ。

問3　親へのポインタをもつ木による集合群(merge find set)の効率的な表現について考える。経路の圧縮は行わないものとし、併合(merge)時はなるべく木の高さが小さくなるように併合するものとする。ただし、木の高さとは、根から末端までのエッジ数の最大値とし、要素数１の時の木の高さは０、要素数２の時の木の高さは常に１であるとする。n は十分に大きい数とする。(各 4 点, 計 12 点)

1)　高さ n の木と高さ n の木を併合(merge)した場合に新しくできる木の高さを答えよ。

2)　同様に高さ n の木と高さ n+2 の木を併合(merge)した場合に新しくできる木の高さを答えよ。

3)　バラバラの n 個の要素を併合して 1 つにしてできた木の高さの最小値と最大値を答えよ。

問4　クイックソートについて以下の問いに答えよ。要素はすべて異なる整数とする。(各 6 点, 計 12 点)

1)　クイックソートについて、最悪の計算量と平均の計算量を答えよ。答えのみでよい。

2)　以下のクイックソートの擬似コードには間違いが 2 つある。間違いを指摘し、どのように直せばよいか説明せよ。

```
QuickSort(array)
    pivot ←　array の最初の要素
    array0 ←　QuickSort（pivot より小さい要素を集めたもの）
    array1 ←　QuickSort（pivot 以上の要素を集めたもの）
    return array0 の後に array1 をつなげたもの
```

3)　上記で修正した擬似コードで示された手順に従って、以下の数列をソートする過程を図示せよ。{36, 95, 56, 84, 34, 96, 24, 66, 32}

問5　基数ソートについて以下の問いに答えよ。（各 6 点,　計 12 点）

1)　以下の整数列を基数ソートで整列する過程を図示せよ。

593, 453, 323, 503, 431, 957,102, 251

2)　基数ソートの計算の手間を見積もれ。ただし，要素数を n，桁数を k，それぞれの桁で取りうる値の種類数を s とする。（例えば上記の例であれば n=8, k=3, s=10 である）

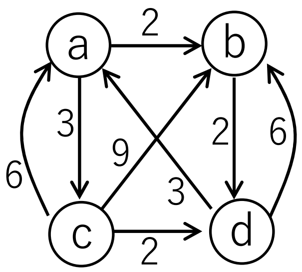
問6　有向グラフにおける全頂点間の最短距離を求める問題について考える。なお、頂点数を n、辺の数を e とする。また、ダイクストラのアルゴリズムには、疎なグラフ(e≈n)に適した実装方法(ヒープによる手法)と、密なグラフ(e≈n^2)に適した実装方法(隣接行列による手法)がある。(各 4 点,　計 16 点)

1)　隣接行列による手法を用いて全頂点間の最短距離を求める場合の計算量を n と e の式として答えよ。

2)　ヒープによる手法を用いて全頂点間の最短距離を求める場合の計算量を n と e の式として答えよ。

3)　フロイドのアルゴリズムを用いて全頂点間の最短距離を求める場合の計算量を答えよ。

4)　以下の有向グラフの全頂点間の最短距離をフロイドのアルゴリズムで求める場合の過程を示せ（途中の計算結果を保持している表の内容がどう変わっていくかを示せ)。



問7　以下は無向グラフ G の最小木を求めるアルゴリズムの概略を示したものである。空欄[i]と[ii]に当てはまるものをそれぞれ答えよ。ただし、グラフ G の頂点集合を V、頂点 u,v 間の辺のコストを d[u,v]（辺がない場合には∞）とする。(各 6 点,　計 12 点)

「任意の 1 点を選び s とする。表 C[ ]を用意し、すべての頂点 v について、C[v] = d[s,v] としておく。未処理頂点の集合 U を用意し、s 以外の点をすべて挿入しておく。表 N[ ]を用意し、すべての頂点 v について N[v]=s としておく。結果を保存する集合 T を用意する。U から C[w]が一番小さい w を選び U から削除し、T に辺{w, N[w]} を加える(a)。U のすべての頂点 v について、条件 [　　　i　　　] が満たされる場合には、以下の処理を行う[　　　ii　　　] (b)。U が空になるまで、(a)~(b)を繰り返す。」

問8　文字列の検索について以下の問いに答えよ。(各 6 点, 計 12 点)

1) KMP 法で検索を行う場合に、以下のパターンに対して生成される表(不一致が見つかったのがパターン中の何文字目かによって、テキスト中におけるパターンの位置をいくつ進めればよいかを決める表)を示せ。

パターン：　PPQPPX

2) BM 法では、パターンの後ろから照合を行っていき、不一致が見つかった時点で以下のような式でテキスト中のパターンの位置 i を更新する(最初、i は 0 から始める)。

i ← i + max(skip[text[i+j]] + j-m+1, 1);

ここで、j は現在照合している文字がパターン中の前から何文字目であるかを示す変数（先頭を 0 とする)、m はパターンの長さ、skip は、効率よく検索を行うためにパターン文字列に基づいてあらかじめ作成しておく表である(不一致が見つかった時のテキスト中の文字の内容(text[i+j])を引数として受け取り、整数を返す)。この時、以下のパターンに対して生成すべき表 skip [ ] の内容を示せ。

パターン：　ACEDCD