問１：

（１）非負整数 $n = 0, 1, 2, \ldots$ に関する関数 $f(n)$ と $g(n)$ に対して，$f(n) = O(g(n))$ であることの定義を書きなさい．ここで，アルファベット大文字の O(オー)はオーダー記法の O である．

（２）非負整数 $n = 0, 1, 2, \ldots$ に関する関数 $f(n)$, $g(n)$ と $h(n)$ に対して，

$$f(n) = O(g(n)) \quad \text{および} \quad g(n) = O(h(n))$$

が成り立つと仮定する．このとき，$f(n) = O(h(n))$ が成り立つことを証明しなさい．

（３）以下の関数をオーダー記法 O (オー) を用いて簡潔に書きなさい．

例： $n^2 + 3n + 100 = O(n^2)$

(i) $n \log n + n^{1.5} + 1000\, n$

(ii) $n! + n^n + n^{10000}$

コメント：

（１）定義を書くだけ.

（２）定義がきちんとわかっていれば，それほど難しくない問題.

（３）極限の計算が出来れば難しくない．大学受験勉強で得た知識を思い出せば出来るでしょう．

問 2：

配列に入れられた整数の集合 3, 2, 0, 5, 8, 3, 4, 1 を，マージソートを使ってソートしたい．

（1）マージソートの基本となるアイディアを，例や図を使いながら説明しなさい．

（2）マージソートでは，ソートされた 2 つの部分配列をマージする必要がある．2 つの部分配列として 0, 2, 3, 5 と 1, 3, 4, 8 を用いて，マージの効率的なやり方について説明しなさい．また，２つの部分配列の長さがそれぞれ $n/2$ のとき，マージの最悪時間計算量を解析しなさい．

（3）ソートすべき整数の数が 2 のべき乗 2^k (k は非負の整数)のとき，マージソートの再帰の深さ（再帰呼び出しの回数）は $O(k)$ となる．これを証明しなさい．

コメント：

（1）説明文としては，以下の内容が書いてあればＯＫ．これに具体例を付ける．

分割統治法に基づいてソートする

n 個の整数の集合をほぼ半分ずつに分割し，それぞれを再帰的にソート．

これにより得られた 2 つのソート列を，マージして一つのソート列を作る．

ポイント：「与えられた配列を 2 分割して，再帰的にソートして，マージする」ということを書けばよい．

細かい説明は必要ない．

（2）前半：アルゴリズムの手順と共に，具体例を書く．

後半：反復回数が $O(n)$であることと，一反復あたり $O(1)$時間が必要なことをきちんと説明してあればよい．

なお，特定のケースについてのみ時間計算量を解析している解答が多く見受けられるが，そのケースがなぜ最悪ケースなのか，説明がないと不十分．

（3）1 回の再帰ごとにソートすべき整数の数が 1/2 になることをふまえると，証明は簡単．

帰納法による証明を使った場合に間違いが多い（オーダー記法の使い方が間違っているため）

問 3：

アルゴリズムを効率的に実行する際，双方向リストは有用なデータ構造である．

（１）双方向リストはどのようなデータ構造であるか？例や図を用いながら詳しく説明しなさい．

（２）現在，双方向リストの中に整数 $5, 8, 2$ がこの順番で格納されていると仮定する．この双方向リストを図で書き表しなさい．

（３）上記で述べた双方向リストの先頭に，整数 3 を挿入したい．その手順について，図を使いながら詳しく説明しなさい．また，その最悪時間計算量についても解析しなさい．

コメント：

（１）複数のセルから構成されていること，各セルに対し，前後のセルへのポインタが与えられていること，さらに先頭と最後尾のセルへのポインタが与えられていること，等を書けばよい．

（３）ポインタをつなぎ替えるときの順番を間違えるとダメ．

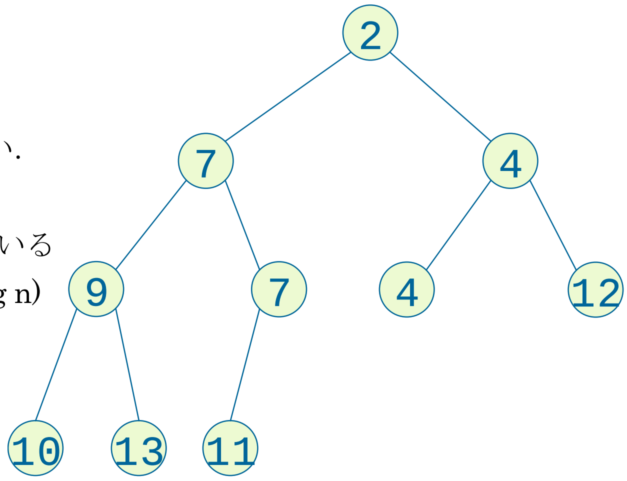
問 4：

右に書いた 2 分木は，ヒープを表している．

（1）2 分木がヒープであるための条件を書きなさい．

（2）右に書いたヒープに，新たに整数 5 を追加したい．その手順を，図を使いながら説明しなさい．

（3）一般に，ヒープが n 個の要素（整数）を持っているとき, 新たな要素の追加に必要な最悪時間計算量は O(log n) となる．これを証明しなさい．



解答欄

（1），（2）は講義資料を参照のこと．

（3）以下の 2 点を証明してあれば十分．

(a) 要素を追加する際には，2 分木の高さに比例した時間を要する．

(b) 要素数 n の 2 分木の高さは，$O(\log_2 n)$である．

問５：

n 個の整数が与えられたとき，第 p 番目に大きい要素（整数）を求めたい．

（１）第 p 番目に大きい要素を求める際，QUICKSELECT と言うアルゴリズムが利用できる．このアルゴリズムの基本的なアイディアについて，例や図を使いながら詳しく説明しなさい．

（２）QUICKSELECT の最悪時間計算量を解析しなさい．

解答欄

（１）以下のことが書いてあればＯＫ．

(i) 軸要素を選ぶ

(ii) 軸要素より大きい要素集合と，軸要素以下の要素集合に分割

(iii) 軸要素より大きい要素数を k とする．

p が k 以下のときは，軸要素より大きい要素集合の第 p 要素を再帰的に求める．

p が k より大きいときは，軸要素以下の要素集合の第 p-k 要素を再帰的に求める．

問題では，「第 p 番目に大きい要素」と書いてありますが，「第 p 番目に小さい要素」を求めるアルゴリズムが書いてあっても可とします．

（２）最悪時間計算量は $O(n^2)$ である．これを証明すればよい．

その証明の際には，再帰の深さが高々 n-1，再帰の各レベルでの計算時間が $O(n)$ であることを理由と共に書いてあればＯＫ．

なお，特定のケースについてのみ時間計算量を解析している解答が多く見受けられるが，そのケースがなぜ最悪ケースなのか，説明がないと不十分．