試験日　　：2025 年 9 月 13 日（土）

入試種別　　：大学院入学試験(4 月入学)修士課程　秋期試験

学部・研究科：先端理工学研究科　数理・情報科学コース

科目名　　：専門科目（C.情報科学）

解答例

C 1

解答で示すプログラムはいずれも C 言語で記述してある.

（1）先頭から末尾まで，v に等しい配列要素の個数を勘定してゆけばよい.

```
int freq(int n, int a[], int v) {
  // O(n)時間アルゴリズム
  // 配列の先頭から，v に等しい要素の個数を勘定してゆく
  int c = 0;
  for (int i = 0; i < n; i++) {
    if (a[i] == v) {
      c++;
    }
  }
  return c;
}
```

（2）配列の先頭から，a[i]の出現回数を順次 freq を呼び出して調べる (i=0,1,2,...)．出現回数の最大値を順次更新して，最後に最大値を返せばよい.

```
int fval(int n, int a[]) {
  // O(n^2)時間アルゴリズム
  int maxc = 0;      //  最大の出現回数（暫定値）
  // 配列の先頭から，a[i]の出現回数を順次調べる (i=0,1,2,...)
  for (int i = 0; i < n; i++) {
    int c = freq(n, a, a[i]); // a[i]の出現回数
    //  これまでの最大の出現回数よりも大きければその回数を記憶
    if (maxc < c) {
```

```
      maxc = c;
    }
  }
  return maxc;  //  最大の出現回数
}
```

（3）ソート済みであれば同じ値は配列 a の連続する要素に並ぶことを利用する．整数変数 i と j を添字として用いる．添字 i=0 から開始して a[i]≠a[j] となる最小の j (> i) を見つける．(ただし，そのような j が存在しない場合には j=n とする)．すると a[i]と同じ値は(a[i]も含めて) j-i 回連続して並んでいるので，この値が a[i] と同じ値の出現回数である．つぎに j の値を i に設定して，同様なことを行う．この繰り返しは i=n になれば終了し，出現回数の最大値を求めればよい．

配列の各要素への参照は定数回であり，実行時間は配列要素数の線形時間である．

C2

（1）

このプログラムは、引数として与えられた整数配列中の要素を大きい順に並び替えるものであるから、この呼び出しから戻った時の d の各要素は次のようになる。

d = { 8, 7, 4, 4, 0 }

（2）

まず、head < tail であれば、配列中の要素の値にかかわらず（☆）の行の文は少なくとも 1 回は実行される。また、quick(d, 0, 4) の場合は、2 つの再帰呼び出しでも、head < j かつ i < tail が成立するため、全体の実行回数の下限は 3 となる。

一方、配列中の要素がすべて相異なっており、すでに大きい順に並んでいる場合は、i = j = ⌊(head + tail) / 2⌋ の時に(☆) の行の文は 1 度だけ実行される。よって、quick(d, 0, 4) の呼び出しでは、i = j = 2 で 1 回実行され、その後、再帰呼び出しの quick(d, 0, 1) と quick(d, 3, 4) について、それぞれ、i = j = 0 と i = j = 3 で 1 回で実行され、全体の実行回数は 3 となる。

以上より、例えば、

```
d = { 4, 3, 2, 1, 0 }
```

の時、実行回数が最小の 3 となる。

（3）

quick(data, head, tail) の呼び出しにおいて、n = tail − head + 1 とする。(☆) の行の実行回数の最大値は、明らかに、n ≦ 1 のときは 0 となり、n = 2 のときは 1 となる。n ≧ 3 の場合については、再帰呼び出しによる分を除いた（☆）の実行回数を m とし、2 つの再帰呼び出し quick(data, head, j) と quick(data, i, tail) について、p = j − head + 1、q = tail − i + 1 とする。n = 3 のときは、(m, p, q) = (1, 1 , 1), (1, 1, 2), (1, 2, 1), (2, 1, 1) の 4 通りがあり、(☆)の実行回数の最大値は 2 となる。同様に、n = 4 のときは、(m, p, q) = (1, 1, 2), (1, 1, 3), (1, 2, 2), (1, 3, 1), (2, 1, 2), (2, 2, 1), (2, 2, 2) の 7 通りがあり、(☆) の実行回数の上限は 4 であり、実際、d のすべての要素が同じ値のときには、(m, p, q) = (2, 2, 2) で 4 となる。さらに、n = 5 のときは、(m, p, q) = (1, 1, 4), (1, 2, 2), (1, 2, 3), (1, 3, 2), (1, 4, 1), (2, 1, 3), (2, 2, 2), (2, 2, 3), (2, 3, 1), (2, 3, 2), (3, 2, 2) の 11 通りがあり、上限は 5 で、d のすべての要素が同じ値のときには、(m, p, q) = (3, 2, 2) で 5 となる。

以上より、quick(d, 0, 4) の呼び出しの場合、例えば、

```
d = { 0, 0, 0, 0, 0 }
```

の時、(☆) の行の実行回数が最大の 5 となる。