# Prob1:

# possible outcome of 4-elements is 4!

In the worst case: # comparisons = height of decision tree that >= ceil(log(4!)) = 5

# Prob2:

static int[] arrange(int[] a) {

int min = a[0];

int max = min;

for (int i = 0; i < a.length; i++) {

if (a[i] < min)

min = a[i];

if (a[i] > max)

max = a[i];

}

if (min < 0) {

int amin = Math.abs(min);

for (int i = 0; i < a.length; i++) {

a[i] += amin;

}

max += amin;

}

int[] bucket = new int[max + 1];

for (int i = 0; i < bucket.length; i++) {

bucket[i] = 0;

} // O(m)

for (int i = 0; i < a.length; i++) {

bucket[a[i]]++;

} // O(n)

System.out.println(Arrays.toString(bucket));

int l = 0, r = bucket.length - 1;

int[] R = new int[a.length];

int k = 0;

while (l < r) {

System.out.println("l:" + l + ", r:" + r);

while (bucket[l] == 0 && l < r)

l++;

if (bucket[l] > 0) {

bucket[l]--;

R[k++] = l;

}

while (bucket[r] == 0 && l < r)

r--;

if (bucket[r] > 0) {

bucket[r]--;

R[k++] = r;

}

} // O(m+n)

if (min < 0) {

int amin = Math.abs(min);

for (int i = 0; i < a.length; i++) {

R[i] -= amin;

} // O(n)

}

return R;

}

# Prob3:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| r[] | 27,72,27 | 1,64 |  |  |  |  |  | 34,16 | 80,8 |
|  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|  |  |  |  |  |  |  |  |  |  |
| q[] | 1,8 | 16 |  | 27,27,34 |  |  |  | 64 | 72,80 |
|  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |

# Prob4:

static Integer find(int[] a) {

int[] bucket = new int[3 \* a.length];

for (int i = 0; i < bucket.length; i++) {

bucket[i] = 0;

}//O(3n)

for (int i = 0; i < a.length; i++) {

bucket[a[i]]++;

}//O(n)

for (int i = 0; i < a.length; i++) {

if (bucket[a[i]] == 1)

return a[i];

}//O(n)

return null;

}