ADA reexam

Mathias Balling Christiansen

December 29, 2024

Contents

1 Opgave

```
bool sumAfToTalLigParameterV1(int arr[], int len, int X) {
  // O(n^2)
  for (int i = 0; i < len - 1; i++) {
    for (int j = 1; j < len; j++) {
      int sum = arr[i] + arr[j];
      if (sum == X) {
        return true;
      }
    }
  }
  return false;
}
bool sumAfToTalLigParameterV2(int arr[], int len, int X) {
  // O(n)
  int right_idx = len - 1;
  int left_idx = 0;
  while (true) {
    int sum = arr[left_idx] + arr[right_idx];
    if (sum == X) {
      return true;
    } else if (sum > X) {
      right_idx--;
    } else if (sum < X) {
      left_idx++;
    }
    if (left_idx == right_idx) {
      return false;
    }
  }
  return false;
}
```

2 Opgave

1. Første loop

```
for (int i = 0; i < 10000; i++) {
    for (int j = 0; j < N*10; j++) {
        for (int k = N; k > 0; k = k/10) {
            x++;
        }
    }
}
```

Ydre loop:

- Løber over i fra 0 til 10000, hvilket giver $O(1)$.

Mellemste loop:

- j løber fra 0 til $10 * N$, hvilket giver $O(N)$

Indre loop:

- k starter ved N og divideres med 10 i hver iteration ($k = k/10$).
- Antallet af iterationer for dette loop er $\log_1 0(N)$, hvilket giver $O(\log N)$.

Samlet tid for de tre loops:

- Den totale tid for denne del af koden er produktet af iterationerne:

$$O(1) \cdot O(N) \cdot O(\log(N)) = O(N \cdot \log(N))$$

3 Opgave

```
const std::string vowels = "aeiouy";
int antalVokaler(std::string str, int l) {
  // Basecase
  if (l < 0) {
    return 0;
  }
  if (vowels.find(str[0]) != std::string::npos) {
    return 1 + antalVokaler(str.substr(1), l - 1);
  } else {
    return antalVokaler(str.substr(1), l - 1);
  }
}
```

4 Opgave

```
void minSortering(int arr[100]) {
  int count_arr[201] = {};
  for (size_t i = 0; i < 100; i++) {
    count_arr[arr[i]]++;
  }
  // 0 -> 200 is always constant O(1)
  for (size_t i = 0; i < 201; i++) {
    // O(N)
    for (size_t j = 0; j < count_arr[i]; j++) {
      std::print("{} ", i);
    }
  }
  std::println("");
}
```

5 Opgave

{0,9,23,106,10,36,38,98,84,12,14,50,55,35,68}

Opfyld max-heap fra bunden.

- Byt 10 og 84
- Byt 50 og 36
- Byt 38 og 55
- Byt 84 og 23
- Byt 106 og 9

- Byt 98 og 9
- Byt 68 og 9

Endeligt array: {0,106,84,98,23,50,55,68,10,12,14,36,38,35,9}

Træet ville være komplet da det er fyldt op fra venstre til højre, men ikke udfyldt helt på sidste niveau for at være perfekt.

6 Opgave

DEMOCRAT:

- D=$11 * 4\%16 = 12$
- E=$11 * 5\%16 = 7$
- M=$11 * 13\%16 = 15$
- O=$11 * 15\%16 = 5$
- C=$11 * 3\%16 = 1$
- R=$11 * 18\%16 = 6$
- A=$11 * 1\%16 = 11$
- T=$11 * 20\%16 = 12$

Index	Value
0	
1	C
2	
3	
4	
5	O
6	R
7	E
8	
9	
10	
11	A
12	D
13	T
14	
15	M

REPUBLICAN:

- R=$11 * 18\%16 = 6$

- E=$11 * 5\%16 = 7$
- P=$11 * 16\%16 = 0$
- U=$11 * 21\%16 = 7$
- B=$11 * 2\%16 = 6$
- L=$11 * 12\%16 = 4$
- I=$11 * 9\%16 = 3$
- C=$11 * 3\%16 = 1$
- A=$11 * 1\%16 = 11$
- N=$11 * 14\%16 = 10$

Index	Value
0	P
1	C
2	
3	I
4	L
5	
6	R
7	E
8	U
9	
10	B
11	A
12	
13	
14	N
15	

7 Opgave

Using Dijkstras Algorithm:

v	Known	d_v	p_v
S	true	0	0
A	true	4	H
B	true	9	F
C	true	16	G
D	true	20	C
E	true	11	H
F	true	7	A
G	true	13	E
H	true	2	S

Jeg bruger Kruskal's. Rækkefølge følger tabellen.

Weight	Node-pair
2	(S,H)
2	(H,A)
2	(H,F)
2	(F,B)
2	(E,G)
3	(G,C)
4	(C,D)
5	(B,E)

Total vægt: 22