

Algorithms - Assignment 1

Καζγκούτης Αθανάσιος Charbel Al Haddad
Παπαδόπουλος Δημήτριος-Λάζαρος

March 24, 2023

Πρόβλημα 1

- Ερώτημα 1

Ο αλγόριθμος που παρατίθεται παρακάτω δέχεται σαν είσοδο μια συστοιχία n στοιχείων όπου n είναι οι ψήφοι -σε ονοματεπώνυμα-μιας κοινότητας. Στόχος του αλγόριθμου είναι:

1. να ελέγξει ποιος/α έχει τους περισσότερους ψήφους.
2. αμα ξεπερνούν ή είναι ίσοι του 50% των συνολικών ψήφων.
3. εφόσον ισχύει το παραπάνω επιστρέψει στην έξοδο το ονοματεπώνυμο του υποψήφιου(ελέγχοντας το ακραίο σενάριο να υπάρχουν 2 υποψήφιοι που ισοβαθμούν ΚΑΙ έχουν το 50% των ψήφων) .

ΑΛΓΟΡΙΘΜΟΣ 1

```
1 function MajorityFinder(A[1...n])
2   majority_person = []
3   maxcount = 0
4   count
5   temp
6   for (i = 1 to n)
7       count = 0
8       temp = A[i]
9       eq=1
10      for (j = 1 to n)
11          if (temp == A[j])
12              count++
13      if (count > maxcount)
14          maxcount = count
15          majority_person[1] = temp
16          if (eq>1)
17              for (u=2 to eq)
18                  majority_person[u] = null
19          eq=1
20          else break
21      else if (count == maxcount){
22          for (u=1 to eq)
23              if (majority_person[u]=temp)
24                  break;
25          else
26              e++
27          majority_person[e] = temp
28      }
29  if (maxcount ≥  $\lceil \frac{n}{2} \rceil$ )
30      return majority_person
31  else
32      return "no person has the majority"
```

ΑΝΑΛΥΣΗ ΑΛΓΟΡΙΘΜΟΥ:

- Στη 1η σειρά ο αλγόριθμος δέχεται τις n ψήφους μέσω μιας συστοιχίας "A[1...n]"
- Στη 2η σειρά ορίζουμε τον πίνακα "majority person" στον οποίο θα αποθηκευτεί το ονόμα του υπερέχον υποψήφιου, το μέγεθος του πίνακα είναι μεταβλητό για να καλύπτει και την περίπτωση της ισοβαθμίας.
- Στις γραμμές 3-5 γίνονται αρχικοποιήσεις που χρησιμεύουν στη καταμέτρηση των ψήφων και των υποψήφιων με τους περισσότερους.
- Στις γραμμές 6-12 ο αλγόριθμος καταμετρεί όλες τις ψήφους. Συγκεκριμένα η αρχική η for χρησιμοποιείται για να προσπελαστούν όλες οι ψήφοι, ενώ η 2η για να γίνει έλεγχος ποιες ψηφοί έχουν το ίδιο ονοματεπώνυμο με την i-στη. Έτσι μέσω το count μετράμε τις συνολικές ψήφους που έχει ο υποψήφιος που βρισκείται στην i-στη θέση του αρχικού πίνακα ενώ στο temp αποθηκεύεται το ονοματεπώνυμο του.
- Στις γραμμές 13-27 γίνεται έλεγχος για να βρέθει ποιος υποψήφιος έχει τις περισσότερες ψήφους, ενώ καλύπτεται η περίπτωση της ισοβαθμίας υποψηφίων μέσω του μεταβλητού πίνακα "majority person"
- Στις τελευταίες γραμμές 29-32 γίνεται έλεγχος αμα κάποιος υποψήφιος έχει \geq του 50% των ψηφών, η εξοδος είτε θα έχει κανένα ένα ή ακόμα και 2 ομνομαα στην ακραία περίπτωση που έχουμε 2 υποψηφίους με τις μισε ψηφους αμφοτεροι.
- Ερώτημα 2

Merge sort

```

1 function mergesort(a[1...n])
2   if (n > 1)
3     return merge(mergesort(a[1...⌊ $\frac{n}{2}$ ⌋]),
4                  mergesort(a[⌊ $\frac{n}{2}$ ⌋ + 1 ... n]))
5   else
6     return a

1 function merge(x[1...k], y[1...l])
2   if (k = 0)
3     return y[1...l]
4   if (l = 0)
5     return x[1...k]
6   if (x[1] ≥ y[1])
7     return x[1] o merge(x[2...k], y[1...l])
8   else
9     return y[1] o merge(x[1...k], y[2...l])

```

```

1 function MajorityFinder2(A[1...n])
2   majority_person = []
3   mergesort(A)
4   for (i = 1 to n)
5     if (A[i] == A[⌈ $\frac{n}{2}$ ⌉ - 1 + i])
6       if (majority_person[1] == null)
7         majority_person[1] = A[i]
8       else
9         majority_person[2] = A[i]
10  return majority_person

```

- Ερώτημα 3

```

1 function MajorityFinder3(A[1...n])
2   majority_person = []
3   HashMap T
4   for (i = 1 to n)
5     if (T.search(A[i]) == true)
6       T[A[i]] = T[A[i]] + 1
7     else
8       T.put([A[i], 1)
9     if (T[A[i]] ≥ ⌈ $\frac{n}{2}$ ⌉)
10      if (majority_person[1] == null)
11        majority_person[1] = A[i]
12      else
13        majority_person[2] = A[i]
14  return majority_person

```

Πρόβλημα 2

- Ερώτημα 1

Algorithm 1

Έστω πίνακας T με στοιχεία n θετικών ακέραιους με εύρος $[0, \dots, k]$ (k ακέραιος)

```
1  for  $i = 0, \dots, k$  do
2       $H[i] = 0$ 
3  end for
4  for  $j = 1, \dots, n$  do
5       $H[T[j]] = H[T[j]] + 1$ 
6  end for
7  for  $i = 1, \dots, k$  do
8       $H[i] = H[i] + H[i - 1]$ 
9  end for
10 for  $j = n, \dots, 1$  do
11      $S[H[T[j]]] = T[j]$ 
12      $H[T[j]] = H[T[j]] - 1$ 
13 end for
```

- Ερώτημα 2