Sveučilište u Splitu
Fakultet elektrotehnike, strojarstva i brodogradnje

Algoritmi

Vježba 7

Nositelj kolegija: izv.prof.dr.sc Matko Šarić
Suradnici u nastavi: asistent Marin Maslov, mag.ing.

Vježba 7

Ova se vježba bavi realizacijom i komparacijom različitih verzija algoritama za *Heap Sort*. Kroz vježbu je potrebno na temelju priloženog pseudokoda i predloška, implementirati iterativni i rekurzivni *Heapify* algoritam. Za dobivene implementacije potrebno je usporediti njihova vremena izvršavanja na nizu različitih veličina.

Pseudokod

```
Heapify(array A, int i, int m)
    l = Left(i)
    r = Right(i)
    max = i

    if(l <= m and A[l] > A[max])
        max = l
    if(r <= m and A[r] > A[max])
        max = r
    if(max != i)
        swap A[i] with A[max]
        Heapify(A, max, m)

Heapify2(array A, int i, int m)
    while(i <= m)
        l = left(i);
        r = right(i)
        max = i

        if(l <= m and A[l] > A[max])
            max = l
        if(r <= m and A[r] > A[max])
            max = r
        if (i != max)
            Swap A[i] with A[max]
            i = max
        else
            break
```

Napomena

```
Left(i) = 2*i
Right(i) = 2*i + 1
```

Predložak za implementaciju (c/c++)

```c
#include<stdio.h>
#include<stdlib.h>
#include<windows.h>
#include<time.h>
#include<conio.h>
#include<iostream>

using namespace std;

int flag;

void Swap(int A[], int i, int j) {
    int temp;
    temp = A[i];
    A[i] = A[j];
    A[j] = temp;
}

void Heapify(int A[], int i, int m) {
    // Implementirati
}
void Heapify2(int A[], int i, int m) {
    // Implementirati
}

void BuildHeap(int n, int A[]) {
    int i;
    for (i = (n - 1) / 2; i >= 0; i--)
    if (flag == 0)
        Heapify(A, i, n - 1);
    else
        Heapify2(A, i, n - 1);
}

void HeapSort(int n, int A[]) {
    int m;
    BuildHeap(n, A);
    m = n - 1;
    while (m >= 1) {
        Swap(A, 0, m);
        m = m - 1;
        Heapify(A, 0, m);
    }
```

```
}

void HeapSort2(int n, int A[]) {
    int m;
    BuildHeap(n, A);
    m = n - 1;
    while (m >= 1) {
        Swap(A, 0, m);
        m = m - 1;
        Heapify2(A, 0, m);
    }
}

int main() {
    srand((unsigned)time(NULL)); //vezivanje rand generatora sa sistemskim
vremenom

    char c;
    int vrijeme1, vrijeme2, vrijeme3, vrijeme4, i, size;

    cout << "Unesi velicinu niza" << endl;
    cin >> size;

    int* A = (int*)malloc(size * sizeof(int));
    int* B = (int*)malloc(size * sizeof(int));

    if (A == NULL || B == NULL) {
        cout << "Nema doboljno memorije za polje ove velicine." << endl;
        return 0;
    }
    else {
        for (i = 0; i < size; i++) {
            A[i] = rand();
            B[i] = A[i];
        }

        flag = 0;
        vrijeme1 = GetTickCount();
        HeapSort(size, A);
        vrijeme2 = GetTickCount();

        cout << "vrijeme Heap Sort sa rekurzivnim Heapify: " << vrijeme2 -
        vrijeme1 << "\n" << endl;

        flag = 1;
```

```
        vrijeme3 = GetTickCount();
        HeapSort2(size, B);
        vrijeme4 = GetTickCount();

        cout << "vrijeme Heap Sort sa iterativnim Heapify: " << vrijeme4 -
        vrijeme3 << "\n" << endl;

        free(A);
        //system("pause");
        return 0;
    }
}
```

Usporedba vremena izvršavanja

n	10	100	1000	10000	100000	1000000
Vrijeme izvršavanja za rekurzivni Heapify [ms]						
Vrijeme izvršavanja za iterativni Heapify [ms]						