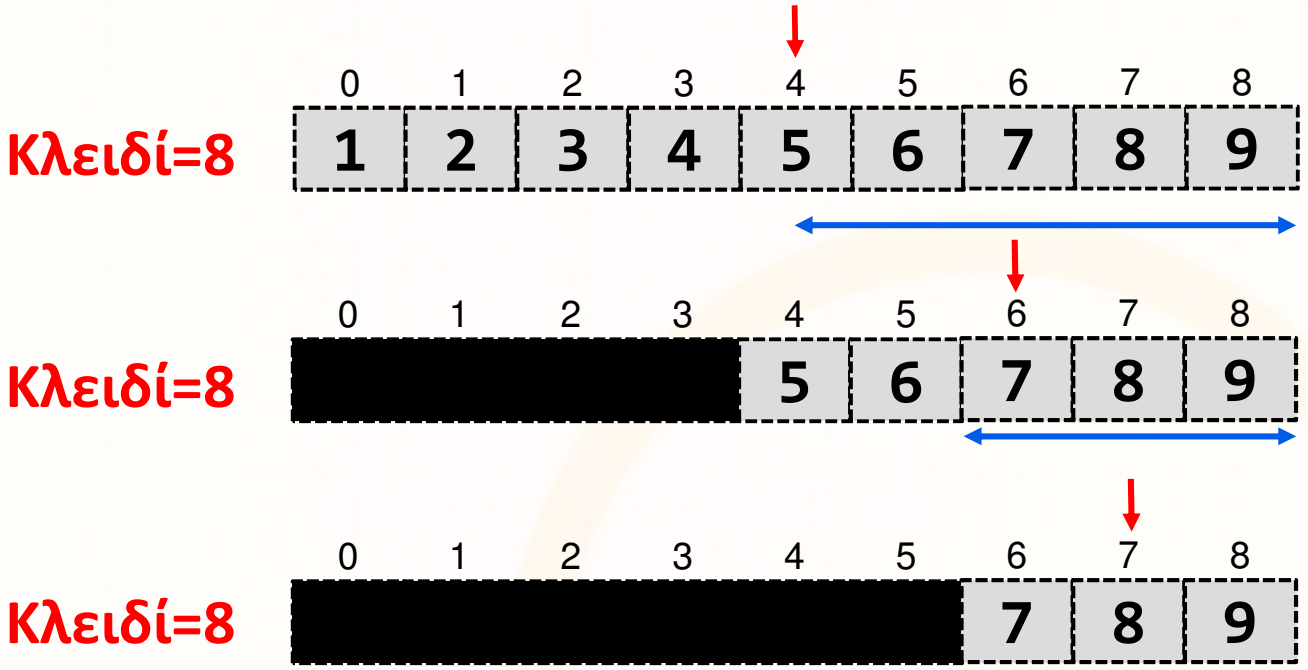
Εργαστήριο 2: Πίνακες

Στην ενότητα αυτή θα μελετηθούν τα εξής επιμέρους θέματα:

- Επεξεργασία Πινάκων
- Υλοποίηση της Δυαδικής Αναζήτησης σε πίνακες
- Υλοποίηση της Ταξινόμησης με Επιλογής σε πίνακες

Δυαδική Αναζήτηση σε πίνακες: BinarySearch

- BinarySearch: ο αλγόριθμος αυτός βρίσκει ένα στοιχείο σε μία ταξινομημένη λίστα ψάχνοντας κάθε φορά στη μέση του πίνακα. Αν το στοιχείο είναι μεγαλύτερο τότε ο πίνακας μοιράζεται στη μέση και χρησιμοποιείται μόνο το δεξί κομμάτι στο επόμενο βήμα.



Δυαδική Αναζήτηση σε πίνακες: BinarySearch (συν.)

Πρότυπο Συνάρτησης

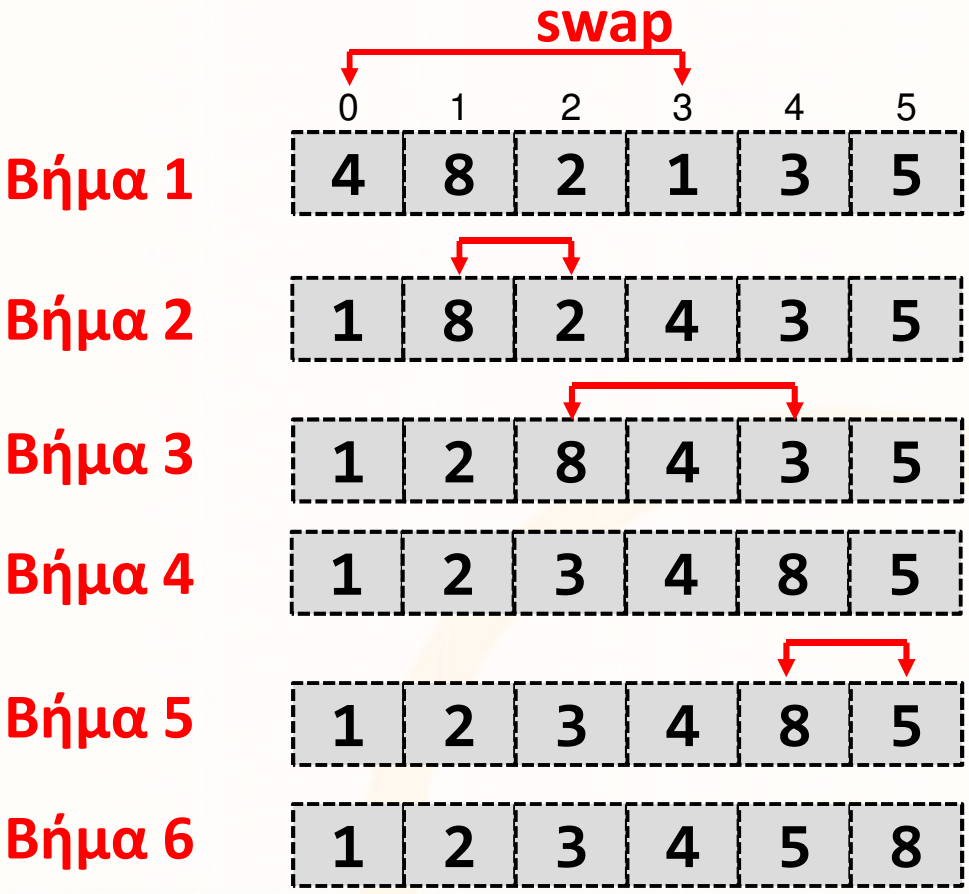
```
int binarySearch(int x[],      //ο πίνακας
                 int length,   //το μέγεθος του x
                 int key)      //το κλειδί αναζήτησης
```

Αλγόριθμος

- Αρχικοποίησε μία μεταβλητή low να δείχνει στην αρχή του πίνακα
- Αρχικοποίησε μία μεταβλητή high να δείχνει στο τέλος του πίνακα
- Με ένα βρόγχο while έλεγχε σε κάθε βήμα αν το high είναι μεγαλύτερο του low και κάνε το εξής:
  - Υπολόγισε τη μέση του πίνακα mid = (low + high) / 2
  - Αν το κλειδί είναι μικρότερο του στοιχείου που βρίσκεται στη θέση mid τότε θέσε σαν high την τιμή mid-1.
  - Αν το κλειδί είναι μεγαλύτερο του στοιχείου που βρίσκεται στη θέση mid τότε θέσε σαν low την τιμή mid+1.
  - Αλλιώς έχεις βρει το κλειδί και επέστρεψε τη θέση του, δηλ. Mid
- Αν τελειώσει ο βρόγχος και δεν έχεις βρει το στοιχείο τότε επέστρεψε - 1

Ταξινόμηση με επιλογή: SelectionSort

- SelectionSort: ο αλγόριθμος αυτός ταξινομεί μία λίστα με στοιχεία. Σε κάθε βήμα i βρίσκει το το i-οστό πιο μικρό στοιχείο και το τοποθετεί στην θέση i.
- Παράδειγμα: `int x = {4, 8, 2, 1, 3, 5};`



Ταξινόμηση με επιλογή: SelectionSort (συν.)

Πρότυπο Συνάρτησης

```
void selectionSort(int x[],     //ο πίνακας
                   int length)  //το μέγεθος του x
```

Αλγόριθμος

- Αρχικοποίησε μία μεταβλητή min_index η οποία θα αποθηκεύει τη θέση του ελάχιστου αριθμού
- Με ένα βρόγχο for που θα ελέγχει ένα ένα τα στοιχεία του πίνακα (i<length) κάνε τα εξής:
  - Θέσε σαν min_index το i
  - Ψάξε ένα-ένα τα στοιχεία με ένα δεύτερο εσωτερικό βρόγχο αρχίζοντας από j=i
    - Αν το στοιχείο που βρίσκεται στη θέση j είναι μικρότερο από το στοιχείο που βρίσκεται στη θέση min_index τότε θέσε min_index = j
  - Με την έξοδο από το βρόγχο, αντάλλαξε τα στοιχεία που βρίσκονται στη θέση i με το στοιχείο που βρίσκεται στη θέση min_index

Υλοποίηση

Υλοποιήστε τις ακόλουθες συναρτήσεις:

- `void printArray(int x[], int length)`
  Τυπώνει ένα πίνακα **x** με μέγεθος **length**

- `int binarySearch(int x[], int length, int key)`
  Εκτελεί δυαδική αναζήτηση για εύρεση του κλειδιού **key** σε ένα πίνακα **x** με μέγεθος **length**

- `void selectionSort(int x[], int length)`
  Ταξινομεί τον πίνακα **x** με μέγεθος **length**

Έλεγχος

Για έλεγχο του προγράμματός σας υλοποιήστε την μέθοδο main ακριβώς όπως φαίνεται πιο κάτω:

```
int main(int argc, char* argv[]) {
  int x[] = {4, 8, 2, 1, 3, 5};
  int length = 6;

  printArray(x,length);    //Τυπώνει 4 8 2 1 3 5
  selectionSort(x,length);
  printArray(x,length);    //Τυπώνει 1 2 3 4 5 8

  printf("%d\n",binarySearch(x, length, 2)); //Τυπ. 1
  printf("%d\n",binarySearch(x, length, 8)); //Τυπ. 5
  printf("%d\n",binarySearch(x, length, 9)); //Τυπ.-1
}
```

Λύσεις

```
void selectionSort(int x[], int length){
  int min_index;
  int tmp;
  for(int i=0; i<length; i++){
    min_index = i;
    for(int j=i+1; j<length; j++){
      if(x[j]<x[min_index])
        min_index=j;
    }
    tmp = x[i];
    x[i] = x[min_index];
    x[min_index]=tmp;
 }
}
```

Λύσεις (συν.)

```
int binarySearch(int x[], int length, int key) {
   int low=0;
   int high = length - 1;
   int mid;
   while( high >= low) {
      mid = (low + high) / 2;
      if( key < x[mid])
         high = mid-1;
      else if (key > x[mid])
         low = mid+1;
      else {
         return mid; break;
      }
   }
   return -1;
}
```

Λύσεις (συν.)

```
void printArray(int x[], int length) {
  for(int i=0; i<length; i++){
    printf("%d ", x[i]);
   }
   printf("\n");
}
```

Λύσεις (συν.)

```
int main(int argc, char* argv[]) {
  int x[] = {4, 8, 2, 1, 3, 5};
  int length = 6;

  printArray(x,length);
  selectionSort(x,length);
  printArray(x,length);

  printf("%d\n", binarySearch(x, length, 2));
  printf("%d\n", binarySearch(x, length, 8));
  printf("%d\n", binarySearch(x, length, 9));

  return 0;
}
```