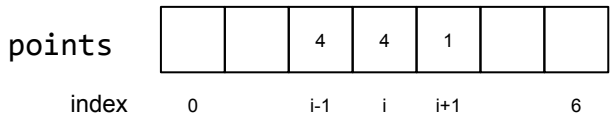
ex4algorithms

Joachim von Hacht

Array som Datastruktur



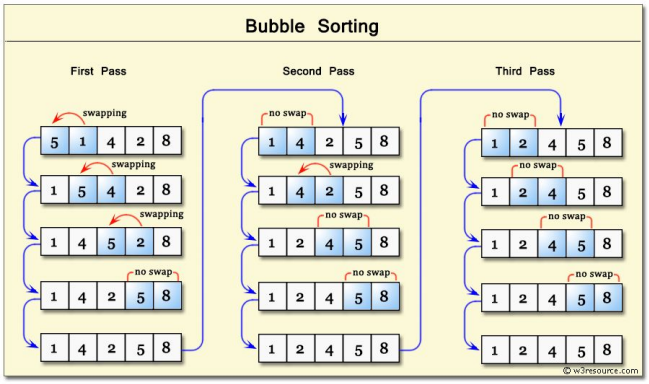

```
int[] points = {0, 0, 0, 0, 0, 0, 0};

int i = 3;
points[i] = 4;            // { 0,0,0,4,0,0,0 }
points[i+1] = 1;          // { 0,0,0,4,1,0,0 }
points[i-1] = points[i]; // { 0,0,4,4,1,0,0 }
```


En array har en struktur, det finns första/sista, före/efter, vänster/höger

- Vi säger att en array är en **datastruktur**.
- Ger oss möjlighet indirekta komma åt variabler: “grannen till”, “tre efter”, “ en före”, …
- Givet ett index i kommer vi åt
  - variabeln till vänster (före) med i-1
  - variabeln till höger (efter) med i+1
  - i-1 eller i+1 får inte hamna utanför array:en, om så: Undantag (som tidigare).

Algoritmer med Arrayer



Definition av Algoritm.

Typiska saker man vill göra med arrayer (och senare matriser): Söka, sortera, flytta/hitta element utifrån visst kriterium, …

- Finns många färdiga lösningar, i form av standardalgoritmer, till dessa problem.
- Bilden: Sorteringsalgoritmen “Bubblesort”

Linjär sökning

```
// This is linear search
int find(int[] arr, int value) {
   for (int i = 0; i < arr.length; i++) {
       if (arr[i] == value) {
           return i;
       }
   }
   return -1;   // Not found!
}
```

Vi returnerar index till ev. funnet värde (inte värdet)

- Gör att vi kan returner ett “omöjligt” indexvärde (-1) om elementet saknas
- Detta är ett standardförfarande.

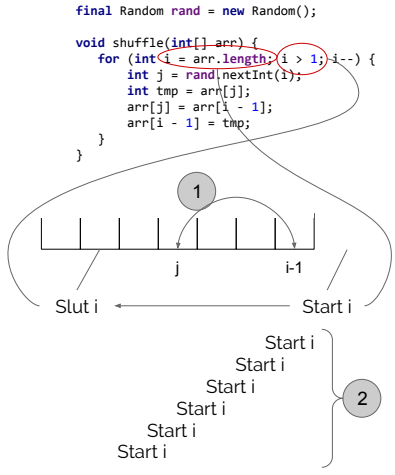
Permutation (Fisher-Yates)

```
final Random rand = new Random();

void shuffle(int[] arr) {
   for (int i = arr.length; i > 1; i--) {
       int j = rand.nextInt(i);
       int tmp = arr[j];
       arr[j] = arr[i - 1];
       arr[i - 1] = tmp;
   }
}
```

För att skapa en godtycklig omordning av elementen i en array (permutation) finns Fisher-Yates algoritm.

Analys Fisher-Yates



Analys inifrån ut:

1. Vad görs i loopen (byter plats på sista och något slumpmässigt element)
2. Vad gör loopen (kortar slumpintervallet, upprepar 1).

Nästlade Loopar

```
// Plot a half square
for (int i = 0; i < 10; i++) {
   for (int j = i; j < 10; j++) {
       out.print("X");
   }
   out.println();
}
```

7

Nästlade loopar mycket vanligt.

- Ofta i samband med algoritmer för arrayer (och matriser, kommer …)
- Ibland styrs den inre loop-variabeln av den yttre (den inre beror på den yttre).

För att förstå nästlade loopar är det vanligen enklast att börja “inifrån”, försök förstå vad som görs i den inre loopen först!

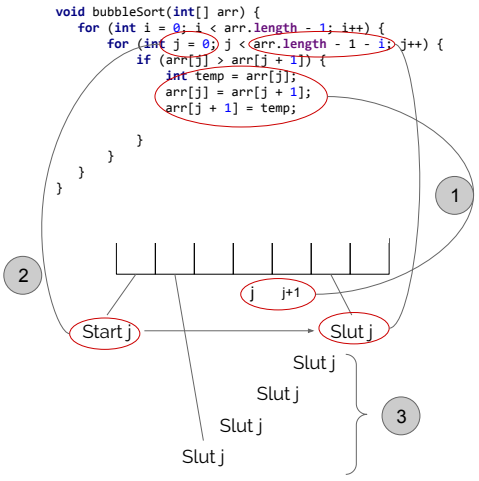
Bubbelsortering

```
void bubbleSort(int[] arr) {
   for (int i = 0; i < arr.length - 1; i++) {
       for (int j = 0; j < arr.length - i - 1; j++) {
           if (arr[j] > arr[j + 1]) {
               int temp = arr[j];
               arr[j] = arr[j + 1];
               arr[j + 1] = temp;

           }
       }
   }
}
```

Sortering av data mycket vanligt. Att sortera en array kan göras med Bubble sort (enkel att förstå men inte så effektiv, undvik)

Analys Bubbelsortering



Börja analys inifrån

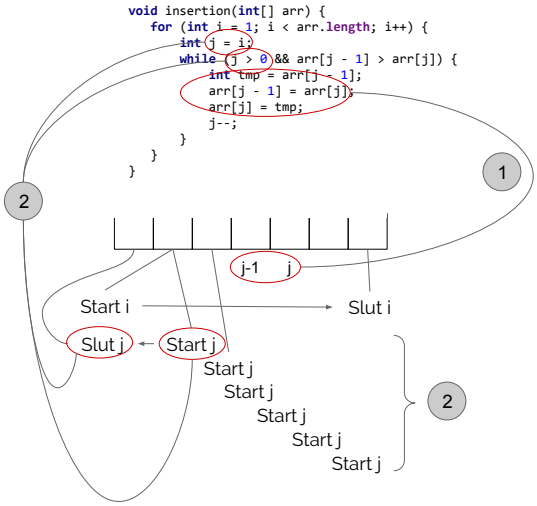
1. Vad görs i innersta loopen (byter plats på j och j+1 om j är större dvs ligger i fel ordning)
2. Var startar j (0 till näst sista)
3. Vad gör yttre loopen (upprepar inre, ändrar slutet på inre loopen)

Insertion sort

```
void insertion(int[] arr) {
   for (int i = 1; i < arr.length; i++) {
       int j = i;
       while (j > 0 && arr[j - 1] > arr[j]) {
           int tmp = arr[j - 1];
           arr[j - 1] = arr[j];
           arr[j] = tmp;
           j--;
       }
   }
}
```

Insertion sort en bättre sorteringsalgoritm.

Analys Insertion sort



Börja analys inifrån

1. Vad görs i innersta loopen (byter plats på j och j-1 om j-1 är större dvs ligger i fel ordning). Flyttar j ett steg till vänster.
2. Var startar j (beror på i, startar längre och längre till höger. Slutar alltid på index 1 eftersom vi har j-1)
3. Vad gör yttre loopen (upprepar inre, ändrar starten på loopen)

Testning av Arrayer

```
int[] arr = { … };


// Simple way to test expected values of array
// Convert to string and compare strings
out.println(Arrays.toString(arr)).equals("[1, 2, 3]"));
```


För att förenkla vid testning kan man omvandla arrayen till en sträng, på samma sätt som vid utskrifter (så slipper vi loopar)

- Nackdel: Det gäller att skriva rätt i det förväntade resultatet!