Stack, 09.11

# Термини

## Buffer overflow

## Overflow

* Push, когато стека е пълен.

## Underflow

* Pop, когато стека е празен.

## Null Elements

* Може да добавяме null елементи.

## Duplicate Elements

* Може да добавяме дублиращи се елементи.

## Limited Data Structure

* Нямаме произволен достъп до различни елементи и всички елементи са с еднакъв приоритет.

## Circular Buffer/King Buffer

## Deque

# Теоретични въпроси

## Въпрос 1

* Възможно ли е при добавянето на елементи в динамичен масив, сложността на разширяване да ни стане не армортизирана константна а линейна.
* Да, ако не използваме рандомизирано разширяване.

## Въпрос 2

* Кога масив за стек, кога свързан списък.

## Въпрос 3

* Stack имплементиран с Динамичен Масив как се представя в паметта
* Като продължителен блок в паметта.

## Въпрос 4

* Възможно ли е да получим StackOverflow ако ползваме свързан списък за имплементацията на stack-a.
* Да.

## Въпрос 5

* Възможно ли е в един масив да симулираме 3 опашки/стека
* Да, може да си разделим масива на 3 части.

## Въпрос 6

* Възможно ли е да използваме стек за опашка и опашка за стек.
* Да.

# Stack - Turing

## Свойства

* Online структура от данни.
* Използва се доста в Операционните Системи.
* Limited Data Structure.
* Initialization с различни имплементации.
* Randomized stack.

## Implementations

* Dynamic array
* Linked List

## Usages

* Undo, redo. Back, forward.
* Reverse word. Hello -> olleH
* Static code analysis (затварящи и отварящи скоби)
* Stack oriented languages.
* Tail recursion optimization.

# Queue

* Многото опашки се ползват за йерархични процеси и синхронни процеси.

# Circular Buffer

# Okasaki Queue

* Queue реализация с 2 едносвързани списъка.
* Добавяш в единия и премахваш от другия и ги.
* Балансираш по време на операции.
* Поддържаме нещо като история.
* По-лесна пойнтър аритметика.