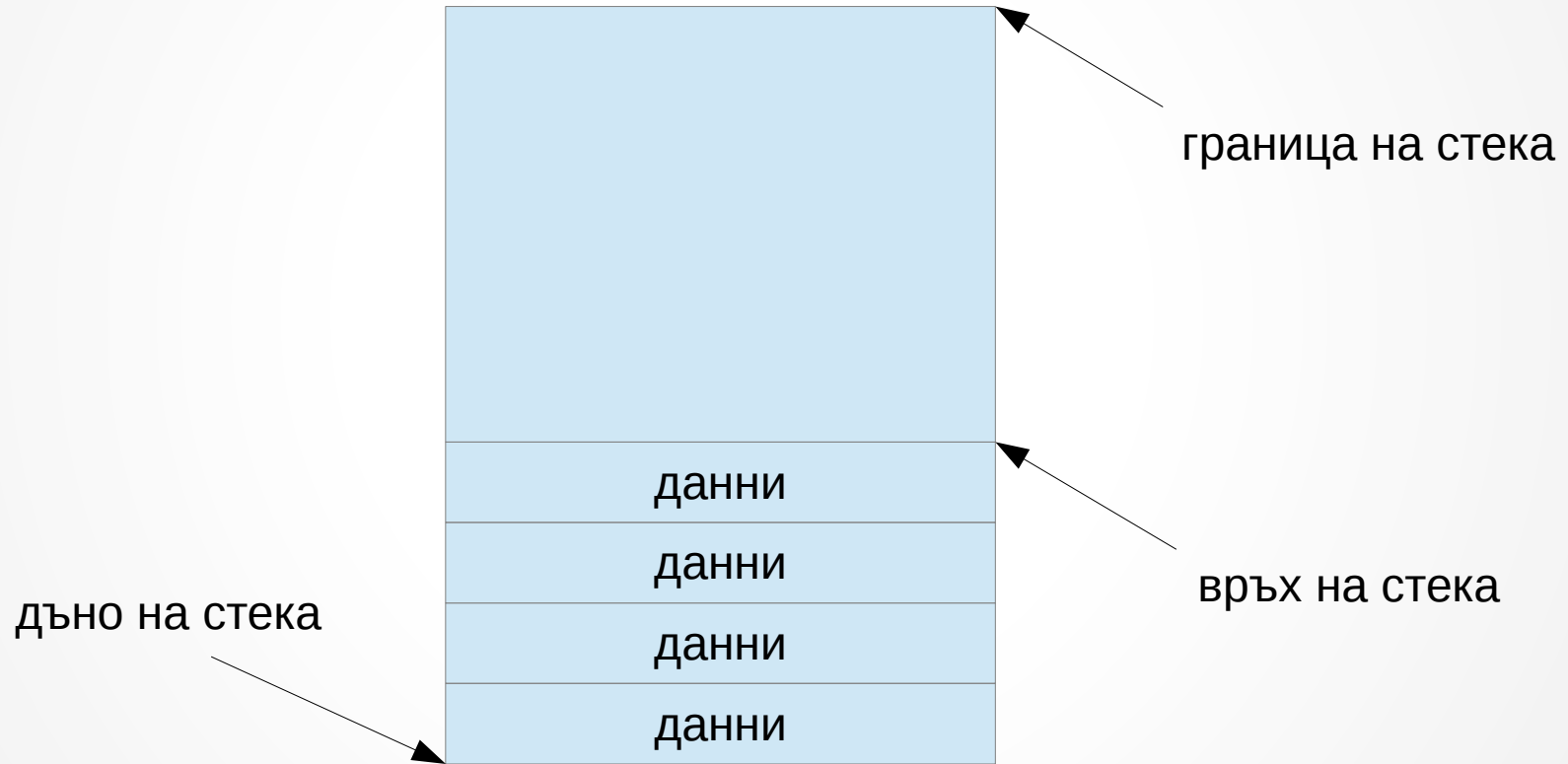


Какво беше това?

данни
данни
данни
данни

Стек

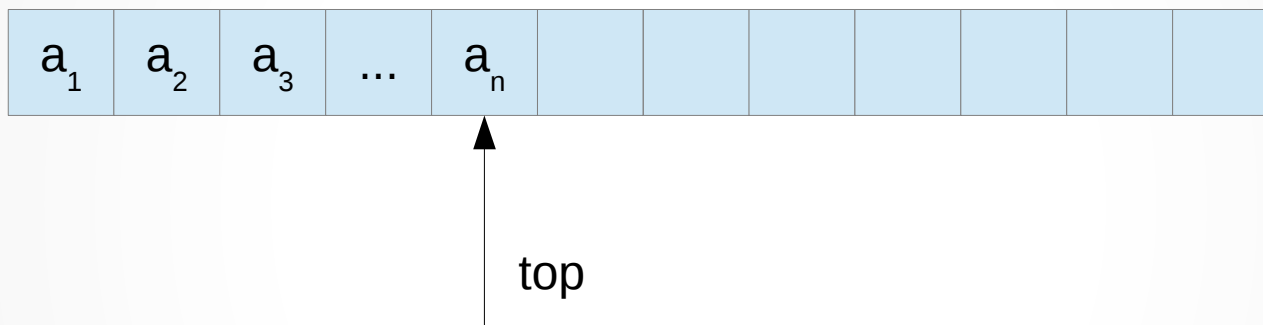
Програмен стек



Стек

- Организация на данни от тип Last In First Out (LIFO)
- Операции:
 - създаване на празен стек (create)
 - проверка за празнота (empty)
 - включване на елемент (push)
 - намиране на последния включен елемент (top)
 - изключване на последния включен елемент (pop)

Последователно представяне на стек



Примерни приложения на стек

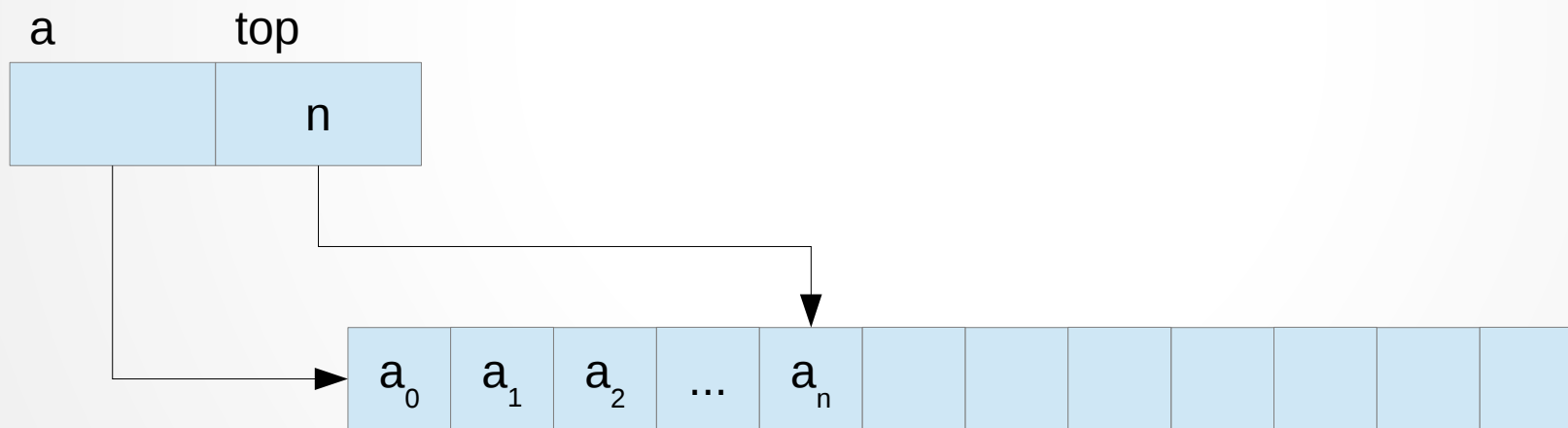
- Намиране на записа на дадено число в k -ична бройна система
- Пресмятане на аритметичен израз
- Проверка за коректност на вложени скоби

Ограничения на стека

- Нашата реализация изисква предварително да зададем горна граница на броя на елементите в стека!
- Ако стекът се препълни, програмата няма да може да продължи да работи... въпреки че компютърът има много налична свободна памет
- Дали е възможно стекът да се „разширява“ при нужда?

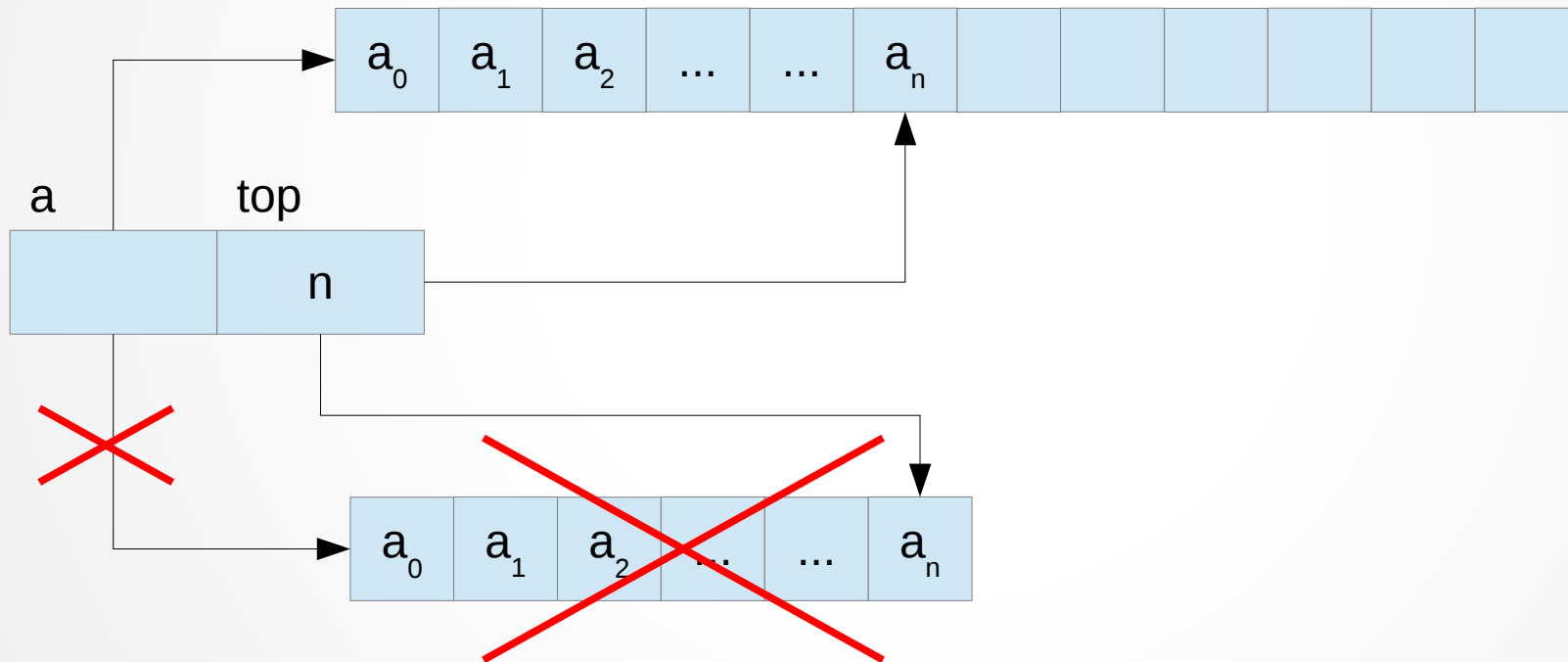
Разширяващ се стек

- Обектът няма да съдържа целия масив
- Ще се пази указател към масив в динамичната памет



Разширяващ се стек

- При нужда стекът ще се разширява

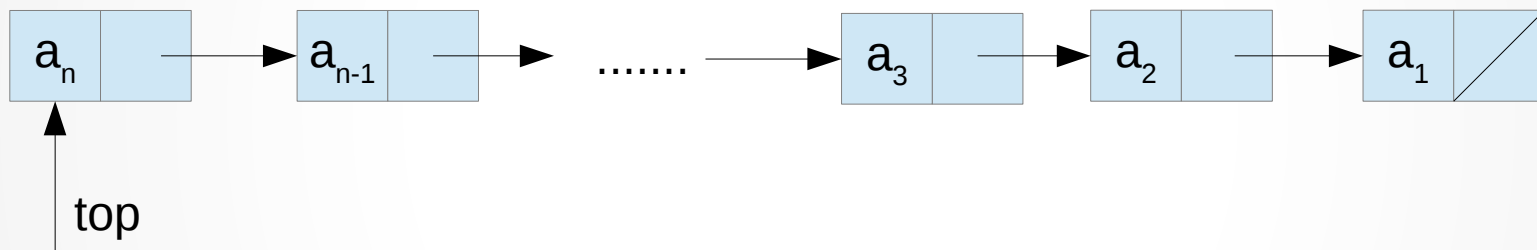


Ограничения на разширяващия се стек

- При разширяване трябва да се копират всички съществуващи данни!
- Операцията `push` обикновено е бърза, но ако се случи да правим разширение може да са доста по-бавни
- Ако стекът се пълни рядко, то в по-голямата част от живота му паметта няма да се използва
- Дали може да се направи стек, при който:
 - не се налага копиране на памет
 - не се държи излишна памет

Свързан стек

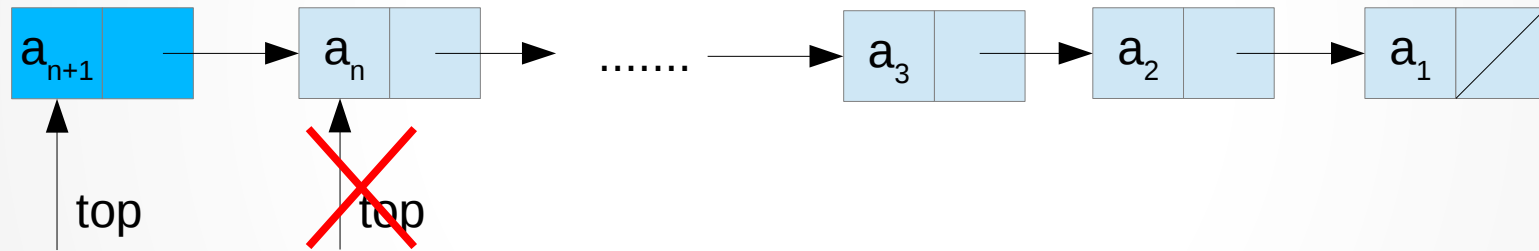
- Представяме стека като „верига“ от двойни кутии



```
struct StackElement {  
    int data;  
    StackElement* next;  
};
```

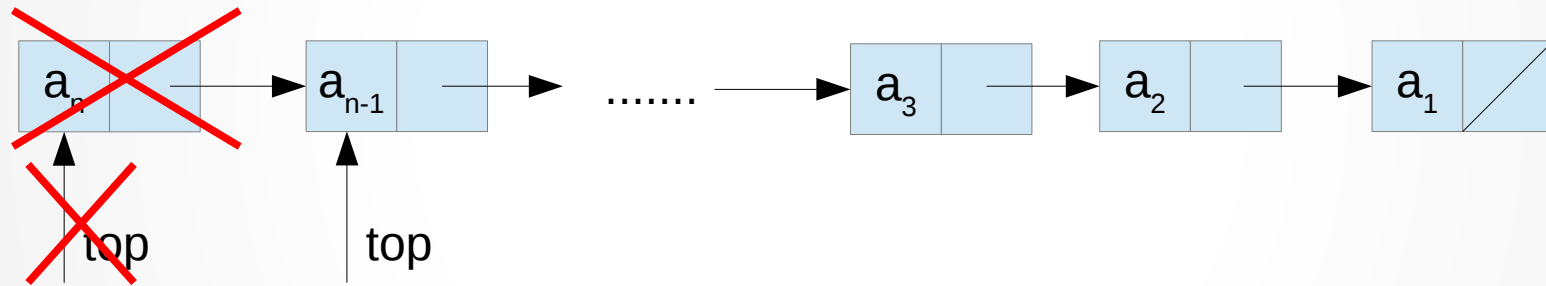
Свързан стек

- Включване на елемент



Свързан стек

- Изключване на елемент



Ограничения на свързан стек

- За всеки елемент се изразходва 2-3 пъти повече памет
- Често се заделят и освобождават малки парчета памет
- Какво се случва при копиране на стек?
 - `LinkedStack s2 = s1; s1.pop(); s2.top(); s2.push(10);`
- Какво се случва при унищожаване на стек?
 - `for(int i = 0; i < 1E8; i++) { LinkedStack s; }`
 - `for(int i = 0; i < 1E8; i++) {
 LinkedStack* s = new LinkedStack;

 delete s;
}`