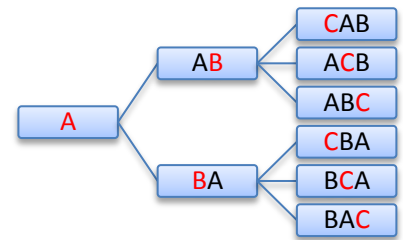


Téma 22: Algoritmy pro generování řetězců a permutací

Permutace n prvků je skupina všech prvků, které jsou uspořádány v jakémkoliv možném pořadí, tzn. výběr prvků závisí na pořadí. Pokud se prvky ve výběru nemohou opakovat, pak počet všech možných výběrů je určen vztahem $P(n) = n!$. Pokud se hovoří o permutacích prvků, jsou tím obvykle myšleny permutace bez opakování.



Příklad: Na vstupu máme řetězec „ABC“. Poté vezmeme písmeno A a k němu přidáme B, jak zleva tak i zprava. Tím vzniknou dvě větve. Poté vezmeme písmeno C, přidáme ho v obou větvích zleva, zprava i doprostřed. Když toto provedeme, vznikne 6 kombinací což je $3!$. Obecně lze říci: (počet prvků)! = počet všech kombinací.

Brutal force prochází již vygenerované číslo a pokud najde vygenerované číslo, generuje dál. Výhodou tohoto postupu je velmi velká rychlost zpočátku, ale na konec pomalé.

Memory force používá pole booleanů a všem nastaví false, když zadá prvek da true když zadá další prvek a už má true, vygeneruje další. Výhodou je, že neprohledává celé pole, a nevýhodou je, že ke konci je již dost pomalý.

Dvouprůchodový jede pořád stejně rychle (nejrychlejší). Jsou 2 průchody. V první průchodu vyplní a-c v dalším průchodu prohází prvky (složitější).

Při generování **řetězců** je potřeba vědět pár pravidel:

- Čísla jsou klíčové znaky... lze jen od 0-9 (nelze tedy napsat 59T ve smyslu padesát devětkrát T).
- Středníky jsou klíčové znaky (nejde je rozšiřovat).
- Musí být také platný vstup. Vnořování je realizováno rekurzí.

Používáme dva jezdce. Pravý se zvyšuje a hledá klíčový znak (0 - 9). Pokud nenajde a je na úrovni nula (není vnořen), tak znak jde do výstupu. Pokud najde a není to středník (je to tedy číslo), a jsme na nulté úrovni tak posuneme levý jezdec na stejnou pozici jako pravý. Zvedneme úroveň. Pokud najdeme středník. Snížíme úroveň a zkontrolujeme, jestli jsme zrovna na úrovni nula. Pokud ano tak vytvoříme for s počtem opakování jako hodnota znaku na pozici levého jezdce (bude zákonitě na čísle). Do výstupu potom uložíme rekurzivní volání s parametrem substringu od levého jezdce+1 (tedy hned za číslem) až po pravý jezdec (tedy ke středníku). Úroveň se mezi rekurzí nepřenáší! Ta je jen pomocná.

Příklad: `2a;` -> vygeneruje `aa`, `32ab,c;` -> vygeneruje `ab ab c ab ab c ab ab c`

Při použití **formatter** je klíčový znak % a některé další znaky. Například „Test %03d“ je decimální číslo na tři místa a volná místa vyplní nulou. To znamená `System.out.format(„Test %03d“, 46)` – vypíše „Test 046“.