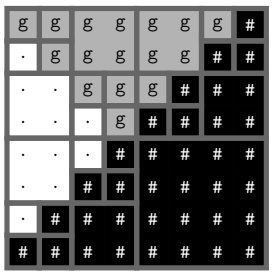
Programovanie (1) v C/C++ 2025/26
Tréningový príklad na skúšku 2a
Quadtree

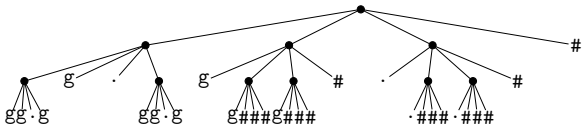
Toto je cvičný príklad pre druhý príklad na skúške. Odporúčame prečítať si zadanie až keď budete úlohu skutočne cvične riešiť.

Kostra na stránke obsahuje implementáciu dátovej štruktúry quadtree, ktorá sa používa v počítačovej grafike. V našom prípade quadtree reprezentuje štvorcový obrázok s $n \times n$ pixelmi. Obrázok obsahuje len málo farieb, ktoré pre jednoduchosť reprezentujeme ako nebiele znaky (obr. (a)).

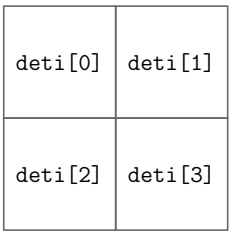
Quadtree je strom, v ktorom je každý vrchol buď list (a teda nemá deti), alebo má presne štyri deti. Každý vrchol zodpovedá určitej štvorcovej ploche vo vstupnom obrázku. Koreň stromu zodpovedá celému obrázku a jeho deti zodpovedajú kvadrantom, ktoré by sme dostali, ak by sme obrázok rozdelili vodorovne aj zvisle na polovicu. Podobne deti každého vnútorného vrcholu zodpovedajú štvorcom, ktoré dostaneme po rozdelení rodičovského štvorca na štyri kvadranty. V delení môžeme pokračovať, kým nedostaneme štvorce obsahujúce iba jeden pixel, čo budú listy, v ktorých máme uloženú farbu príslušného pixelu. List však môže zodpovedať aj väčšiemu štvorcu, pokiaľ je tento štvorec v obrázku celý vyplnený rovnakou farbou (obr. (a), (b)).



(a) Príklad obrázku 8×8 pixelov s tromi farbami, pričom znak `'.'` zastupuje bielu, `'#'` čiernu a písmeno `'g'` sivú. Orámované štvorce zodpovedajú listom v strome.



(b) Strom pre obrázok v časti (a). V listoch je zobrazený uložený znak reprezentujúci farbu daného štvorca.



(c) Poradie detí pre jednotlivé kvadranty v poli `deti` štruktúry `vrchol`. Napríklad `deti[0]` je ľavý horný kvadrant.

Typ `vrchol` reprezentuje vrchol stromu a obsahuje nasledujúce zložky. Premenná `jeList` určuje, či vrchol je listom. Znak `farba` v liste obsahuje farbu daného štvorca a pre vnútorné vrcholy obsahuje 0. Čísla `riadok` a `stlpec` sú súradnice ľavého horného rohu štvorca pre tento vrchol, pričom riadky a stĺpce číslujeme $0, \ldots, n-1$ zhora nadol, resp. zľava doprava. Kladné celé číslo `velkost` určuje počet riadkov (a teda aj stĺpcov) vo štvorci. Štvorprvkové pole `deti` obsahuje smerníky na vrcholy detí, pričom v liste sú nastavené na `NULL`. Poradie detí v poli je podľa obr. (c). Vašou úlohou je naprogramovať nasledujúce tri funkcie:

(1) Funkcia `najdiFarbu` dostane vrchol v a súradnice riadku a stĺpca a vráti farbu (t.j. znak), nachádzajúci sa na daných súradniciach. Vo funkcii môžete použiť už hotovú funkciu `jeVnutri`, ktorá zistí, či dané súradnice sú vo vnútri štvorca pre zadaný vrchol.

(2) Funkcia `zjednodus` dostane koreň stromu a strom zjednoduší tak, aby predstavoval ten istý obrázok, ale aby mal čo najmenej vrcholov. Teda ak nejaký vrchol, ktorý nie je list, predstavuje jednofarebnú plochu, nahradíme ho listom s danou farbou. Odporúčame použiť už hotovú funkciu `daSaZjednodusit`, ktorá pre daný vrchol zistí, či je možné ho zjednodušiť na jediný list. Pozor, táto funkcia predpokladá, že deti zadaného vrchola sú už zjednodušené. Všetky vrcholy, ktoré zo stromu zrušíte, odalokujte.

(3) Funkcia `prevrat` dostane koreň stromu a zmení strom tak, aby zodpovedal obrázku, v ktorom je otočené poradie riadkov, t.j. prvý riadok sa stane posledným, druhý predposledným atď. V strome nevytvárajte ani nemažte vrcholy, iba vhodne presúvajte položky poľa `deti`. Súradnice riadkov a stĺpcov uložené vo vrcholoch nemusíte meniť, tie sa dopočítajú vo funkcii `main` pomocou funkcie `suradnice`. Vo funkcii `prevrat` sa vám môže zísť funkcia `swap`, ktorá vymení dva smerníky na vrchol.

Podľa potreby môžete naprogramovať a použiť aj ďalšie funkcie, ale nemeňte už hotové časti programu. Dodržujte popisy funkcií a význam položiek štruktúry `vrchol` uvedené v zadaní a v kostre.

V programe je už hotové načítanie dát. Na prvom riadku vstupu je veľkosť obrázka n, čo bude vždy mocnina 2 (1, 2, 4, 8,...), aby sa ľahko delili štvorce na rovnako veľké kvadranty. Nasleduje obrázok ako n riadkov, každý s n nebielymi znakmi. Hotový je aj výpis obrázku, potrebuje však vašu funkciu `najdiFarbu`. Pod obrázkom sa vždy uvedie počet listov v strome. Funkcia `main` vypíše najskôr načítaný strom, v ktorom všetky listy zodpovedajú jednému pixelu, potom ho zjednoduší vašou funkciou a opäť vypíše a nakoniec strom prevráti a vypíše.

Príklad vstupu:

```
8
ggggggg#
.ggggg##
..ggg###
...g####
...#####
..######
.#######
########
```

Príklad výstupu:

```
Vstupny strom:
ggggggg#
.ggggg##
..ggg###
...g####
...#####
..######
.#######
########
Pocet listov 64

Strom po zjednoduseni:
ggggggg#
.ggggg##
..ggg###
...g####
...#####
..######
.#######
########
Pocet listov 31

Strom po prevrateni:
########
.#######
..######
...#####
...g####
..ggg###
.ggggg##
ggggggg#
Pocet listov 31
```