

Dragon Tree Fields

Abychom spočetli kvality všech polí, připravíme si pomocné pole, do něhož předpočítáme vhodné částečné součty. Kvalitu kteréhokoliv pole pak spočteme jako kombinaci hodnot pouze čtyř prvků pomocného pole. Na vedlejším obrázku je znázorněna myšlenka tohoto triku.

V horní části je původní pole, pod ním je pole pomocné.

Pomocné pole registruje na pozici (r, s) součet všech prvků původního pole které se nacházejí v obdélníku, jehož levý horní roh je vždy v levém horním rohu pole a pravý dolní roh na pozici (r, s) . Kupříkladu růžová 101 v pomocném poli je součet hodnot 31, 2, 12, 26, 13, 17 v původním poli.

Když nyní například chceme spočítat součet všech prvků ve fialové ohraničené oblasti (15, 16, 14, 28, 27, 11) původního pole, uděláme to tak, že vezmeme součet všech prvků v oranžově(žlutě) ohraničené oblasti a od něj odečteme součet všech prvků jednak v modře a také v zeleně ohraničené oblasti. Jenže ouha, čísla v růžově ohraničené oblasti jsme tak vlastně odečetli dvakrát, takže je musíme znova jednou přičíst. To je zapsáno úplně dole na obrázku.

Nevadí, že oblast v ukázce je obdélníková, pro čtvercové oblasti bude platit identický postup. Například součet čísel v centrální čtvercové oblasti o rozměru 2×2 (s hodnotami 21, 22, 15, 16) získáme výpočtem

$$278 - 148 - 127 + 71 = 349 - 275 = 74. \quad (\text{Kontrola: } 21 + 22 + 15 + 16 = 74.)$$

Zbývá vymyslet efektivní výpočet pomocného pole. Ten je založen na analogickém nápadu. Pole vyplníme jedním průchodem postupně po řádcích. Na pozici (r, s) v pomocném poli napřed dáme hodnotu původního pole na téže pozici (r, s) a k ní přičteme již předpočítané součty z pomocného pole na pozicích $(r-1, s)$ a $(r, s-1)$ a nezapomeneme odečíst součet na pozici $(r-1, s-1)$ v pomocném poli. Např. v pomocném poli vypočteme $135 = 21 + 84 + 101 - 71$ apod.

Pro implementační zjednodušení, aby se snáz stejným postupem počítaly i hodnoty v prvním řádku a sloupci pomocného pole, můžeme si představovat, že obě pole mají ještě nahoře a vlevo jeden umělý řádek a sloupec vyplněné nulami.

Poslední vylepšení spočívá v tom, že vlastně nepotřebujeme původní pole, protože nás v úloze zajímají jen součty, a ty jsou zaneseny do pomocného pole. Hodnoty původního pole načteme ze vstupu rovnou do pole pomocného a po kompletním načtení přepočítáme hodnoty v pomocném poli postupem z předchozího odstavce.

V konečné fázi řešení pak jen projdeme (třemi vloženými cykly) všechny možné čtvercové oblasti v pomocném poli, u každé výše naznačeným postupem spočteme její součet prvků a ty oblasti, které vyhovují podmínce úlohy (leží v intervalu $[Q1, Q2]$), si uložíme do seznamu. Tento seznam oblastí posléze seřadíme podle jejich kvality (= součtu prvků). Seřazený seznam projdeme dvěma indexy, tedy dvěma cykly. První index prochází zleva doprava, druhý index ve vnitřním cyklu začíná vpravo od aktuální pozice prvního indexu a zastaví se pokaždé v okamžiku, kdy již rozdíl kvalit oblasti na pozici prvního indexu a na pozici druhého indexu je větší než dosud spočtený minimální rozdíl. Přitom je nutno kontrolovat, zda se oblasti na pozicích prvního a druhého indexu nepřekrývají, ale to je již snadné cvičení, na to stačí vyhodnotit nejvýše 4 nerovnosti, viz kód.

1.

31	2	4	33	5	36
12	26	9	10	29	25
13	17	21	22	20	18
24	23	15	16	14	19
30	8	28	27	11	7
1	35	34	3	32	6

2.

31	33	37	70	75	111
43	71	84	127	161	222
56	101	135	200	254	333
80	148	197	278	346	444
110	186	263	371	450	555
111	222	333	444	555	666

$$15 + 16 + 14 + 28 + 27 + 11 = 101 + 450 - 254 - 186 = 111$$