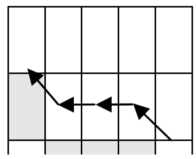
1.
Následující matice 5x4 polí reprezentuje výřez v obrázku. Každé políčko představuje jeden pixel s uvedenou hodnotou. Postupovat lze jen zleva doprava, a to vodorovně, nebo pod úhlem ±45 stupňů, tj. do políčka s indexem [i,j] se lze dostat z políčka v předchozím sloupci s indexem [i-1, j-1], [i, j-1], [i+1, j-1], kde i=řádkový index a j= sloupcový index.

j →

i	2	12	23	18
|	23	23	6	12
|	20	9	12	10
v	18	15	11	8
	16	12	8	7



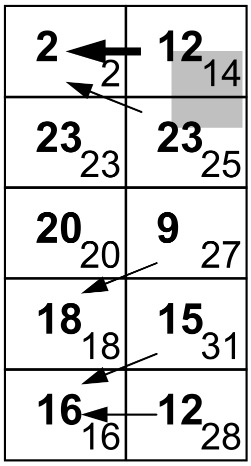
Dynamickým programováním nalezněte spojnici levé a pravé strany (touto spojnicí se míní posloupnost políček, jedno políčko na sloupec, začínající ve sloupci s indexem j=1 a končící ve sloupci s j=4) takovou, že součet hodnot podél ní je minimální.

Jaké mezivýsledky si musíte pamatovat pro jednotlivé sloupce a pro celou tabulku?

Myšlenka řešení:

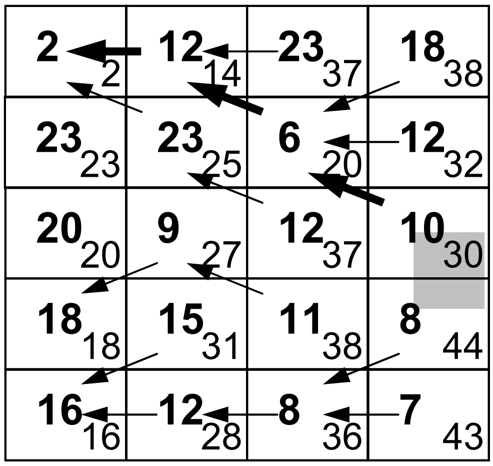
Představme si, že máme matici M obsahující pouze dva sloupce. Abychom našli uvedenou spojnici, uděláme toto: Probereme postupně všechny prvky ve druhém sloupci a pro každý prvek M[i,2] zaregistrujeme minumum ze tří součtů: S[i,2] = min {M[i-1,1]+M[i,2], M[i,1]+M[i,2], M[i+1,1]+M[i,2]}, tedy součtu hodnoty M[i,2] a některého z jeho možných tří předchůdců. Pro toto minimum zaregistrujeme také, pro kterého z předchůdců nastává (např, pomocí hodnot –1, 0, +1).

Když toto provedeme, stačí jen projít znova druhý sloupec a najít nejmenší hodnotu součtů S[i,2]. Protože je tam zaregistrován i přechůdce, je kompletní (byť krátká) hledaná cesta nyní známa. Podívejte se na následující obrázek. Velká čísla představují původní hodnoty M[i,j], malá čísla představují součty podél nalezených spojnic končících v jednotlivých prvcích druhého sloupce, tj hodnoty S[i,j] (pro zjednodušení případné implementace položme S[i,1] = M[i,1] pro všechna i). Šedě je podbarveno celkové minimum.



Nyní si představme, že k tomuto řešení připojíme dále třetí sloupec. Ve druhém sloupci je pro každý jeho prvek známa optimální spojnice, která končí v tomto prvku. Nyní tedy stačí projít třetí sloupec stejně jako jsme poprvé procházeli druhý sloupec a zaregistrovat pokaždé minimum S[i,3] = min {S[i-1,2]+M[i,3], S[i,2]+M[i,3], S[i+1,2]+M[i,3]} a zaregistrovat příslušného předchůdce.

Přidávání dalších (čtvrtého, pátého, atd...) sloupců se řídí stejným pravidlem, takže v dané úloze získáváme následující definitivní obrázek



Celkem tedy si pro každý sloupec j musíme pamatovat hodnoty S[i,j] i=1..5, a předchůdce každého prvku. Předchůdce můžeme ukládat do další matice P, kde P[i,j] je –1, 0 nebo +1.

2.

Následující matice *m* reprezentuje výřez v obrázku. Každé políčko představuje jeden pixel s uvedenou hodnotou. Postupovat lze jen zleva doprava, a to vodorovně, nebo pod úhlem ±45 stupňů, tj. do políčka s indexem [i,j] se lze dostat z políčka v předchozím sloupci s indexem [i-1, j-1], [i, j-1], [i+1, j-1], kde i=řádkový index a j= sloupcový index.

	j →			
i	3	12	23	18
|	15	23	6	12
|	20	9	12	10
v	18	14	19	8
	16	12	8	7

Dynamickým programováním nalezněte spojnici levé a pravé strany (touto spojnicí se míní posloupnost políček, jedno políčko na sloupec, začínající ve sloupci s indexem j=1 a končící ve sloupci s j=4) takovou, že součet absolutních rozdílů hodnot podél cesty je minimální. Rozdílem hodnot podél cesty je např. pro políčko [i, j] hodnota abs(m[i,j] – m[i_{prev},j-1]), kde i_{prev} nabývá jedné z hodnot {i-1, i, i+1}.
Jaké mezivýsledky si musíte pamatovat pro jednotlivé sloupce a pro celou tabulku?

Myšlenka řešení je zcela identická jako v předchozí úloze (v sousedním oddělení). Místo pouhých součtů podél spojnice budeme registrovat součty absolutních hodnot rozdílů, jinak se nezmění vůbec nic.

Matici S tedy budeme počítat ze vztahu:

S[i,j] = min {S[i–1, j–1] + abs(M[i, j] – M[i–1, j–1]),

S[i, j–1] + abs(M[i, j] – M[i, j–1]),

S[i+1, j–1] + abs(M[i, j] – M[i+1, j–1]) }

Výslednou matici tu pro nedostatek času neuvádím, spočtěte si ji jako snadné cvičení.

3.

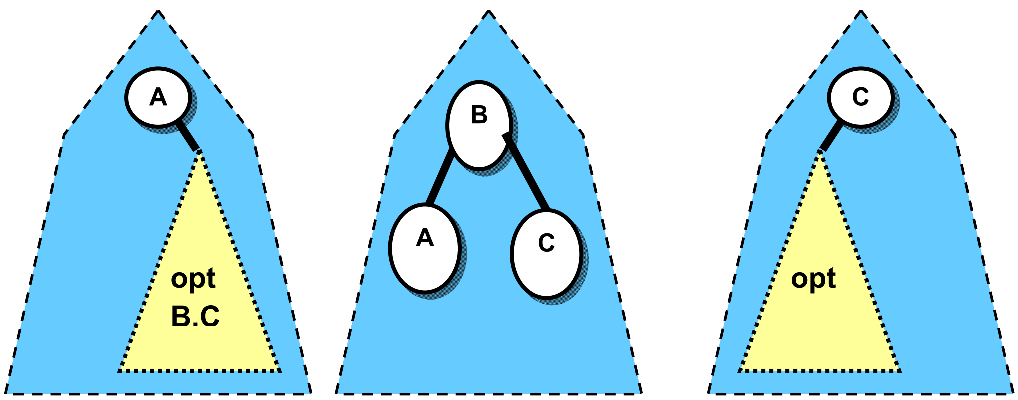
Jsou dány prvky s klíči A-G. Pravděpodobnost dotazu na jednotlivé konkrétní klíče je dána níže uvedenou tabulkou. Metodou dynamického programování sestrojte optimální strom z hlediska operace FIND, tj, najděte takový strom, v němž průměrný počet dotazů na hodnotu klíče během jedné operace FIND je nejmenší (Předpokládáme, že obsah stromu se dlouhodobě nemění). Napište, jak vypadá tabulka ohodnocení optimálních podstromů a tabulka jejich kořenů a výsledný strom namalujte.

A: 0.10
B: 0.10
C: 0.25
D: 0.35
E: 0.10
F: 0.05
G: 0.05

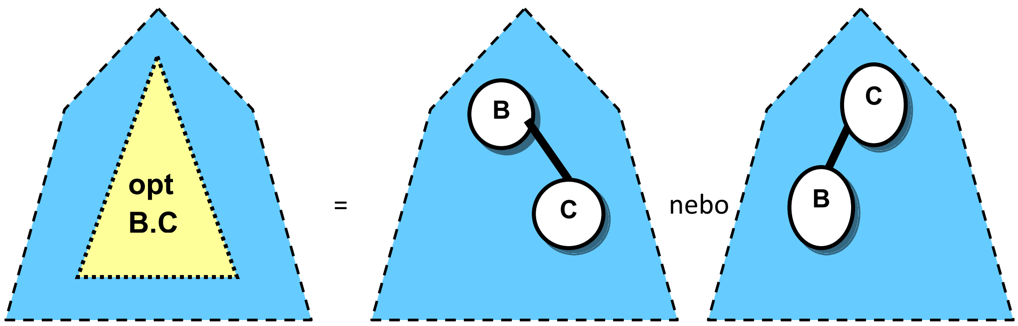
Řešení: Nejlepší je se podívat do přednášky, kde je vše pěkně popsáno a definováno. Zde jen několik „praktických“ poznámek.
Tabulka ohodnocení optimálních podstromů *t:*

- Má stejný počet řádek, jako je počet uzlů. Řádkový index odpovídá počátečnímu uzlu *L* z intervalu uzlů *L..R,*
- má o jeden sloupec více, než je uzlů. Sloupcový index odpovídá koncovému uzlu *R* z intervalu uzlů *L..R*,.
- je dobré si uvědomit, že uzly mají pevně dané pořadí (jejich klíče jsou uspořádány)
- má tedy pod diagonálou samé nuly, protože podstrom s L < R nemá smysl,
- vše pod diagonálou chápeme jako prázdný podstrom,
- na diagonále jsou hodnoty „jednouzlových“ podstromů, tedy přímo pravděpodobnost dotazu (1-krát, neboť hloubka je 1. V předchozích přednáškách byla hloubka stromu s jedním uzlem definována jako rovna nule. Uváděním hloubky od jedné vznikl trochu zmatek. Konzistentnější by bylo ponechat hloubku stromu s jedním uzlem rovnu nule a uvádět, že hodnotu pravděpodobnosti dotazu uzlu vynásobíme hloubkou uzlu zvětšenou o 1.),

- Při vyplňování tabulky musíme do políčka o souřadnicích *t*[L,R] vyzkoušet všechny možné polohy kořene v rámci intervalu L..R. Pro L=A a L = C jsou možnosti následující:



Opt B,C znamená optimální podstrom sestávající z uzlů B a C, tj. výsledek předchozího kroku dynamického programování. Je to ten, jehož ohodnocení bylo menší, v našem případě podstrom s kořenem C.



		A	B	C	D	E	F	G
A	0	0.1	0.3	0.75	1.45	1.75	1.9	2.1
B	0	0	0.1	0.45	1.15	2.03	1.5	1.7
C	0	0		0.25	0.85	1.05	1.2	1.4
D	0	0			0.35	0.55	0.7	0.9
E	0	0				0.1	0.2	0.35
F	0	0					0.05	0.15
G	0	0	0					0.05

		A	B	C	D	E	F	G
A	-	A	AB	C	C	CD	D	D
B	-	-	B	C	CD	D	D	D
C	-	-		C	D	D	D	D
D	-	-			D	D	D	D
E	-	-				E	E	EFG
F	-	-					F	FG
G	-	-						G

Zažlucená políčka jsou nejednoznačná, protože vyjde stejná hodnota pro více různých kořenů.

Např. hodnota pro podstrom BCD na pozici t[B,D] se vypočítá jako:
0.1+0.25+0.35 + min(0+0.85, 0.1+0.35, 0.45+0) = 0.7 + min(0.85, 0.45, 0.45) = 1.15

Výsledný strom tedy může vypadat například takto: