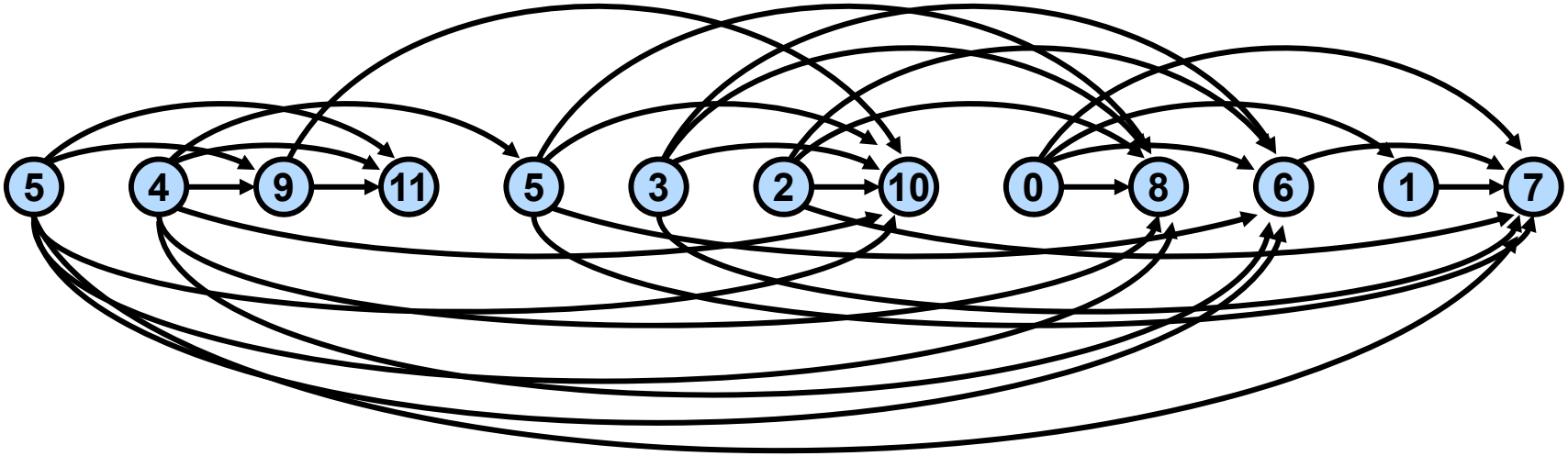
Nejdelší rostoucí podposloupnost

Koncepční přístup I

Transformace na známou úlohu

Prvky posloupnosti budou uzly DAG, který je již topologicky uspořádán, pořadí v posloupnosti = pořadí v top. uspořádání.
Hrana x --> y existuje právě tehdy,
když x je v posloupnosti dříve než y a navíc x<y.

V tomto DAG hledáme nejdelší cestu.



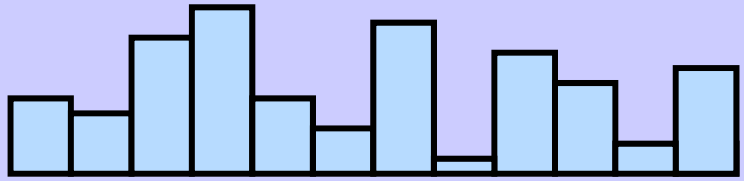
Algoritmus je znám, má složitost $\Theta(N+M)$, tedy $O(N^2)$.
Např. pro rostoucí posloupnost má složitost $\Theta(N^2)$.

Nejdelší rostoucí podposloupnost

Koncepční přístup II: **Sestav nezávislý a rychlejší algoritmus:**

Registrujme optimální podposloupnosti (OPP) všech možných délek.
OPP každé délky má co nejmenší možný poslední prvek.
To ji dává potenciál budoucího růstu.
Postupně metodou DP aktualizujme tyto optimální podposloupnosti.

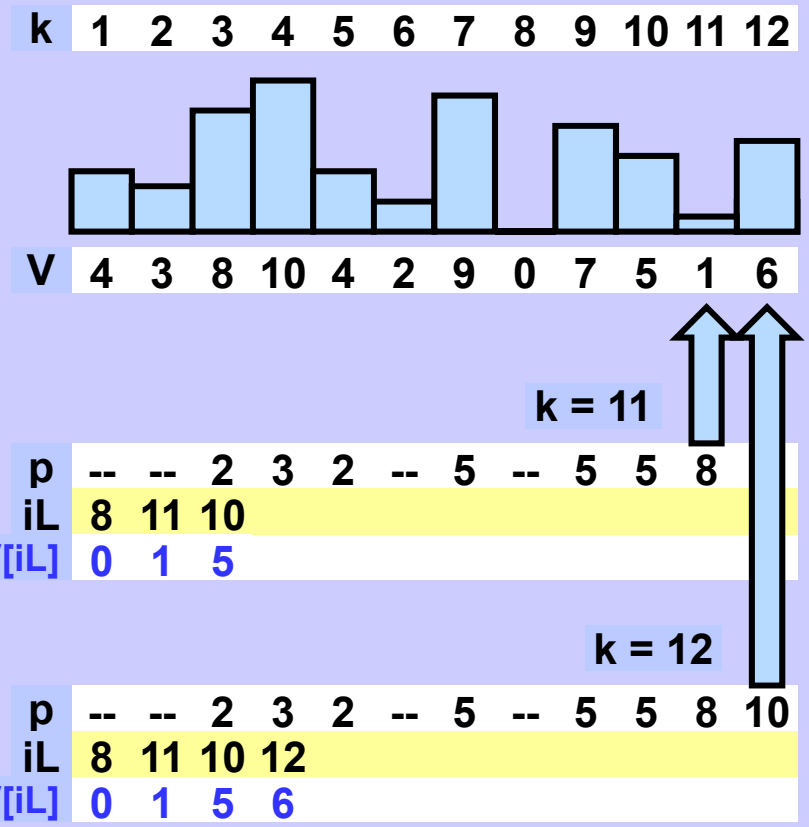
k	1	2	3	4	5	6	7	8	9	10	11	12
V	5	4	9	11	5	3	10	1	8	6	2	7
p	--											
iL	--											



k .. index prvku
V[k] .. hodnota prvku
p[k] .. index předchůdce prvku
iL[d] .. index posledního prvku v rostoucí podposloupnosti délky d = 1, 2, ..., N.

Pro každý index k:
Nechť d je index <u>největšího</u> prvku, pro který platí V[iL[d]] < V[k].
Potom iL[d+1] := k, p[k] = iL[d], pokud d existuje.
Jinak iL[1] := k, p[k] = null.
V[iL[d]], d = 1..N je neklesající, lze v ní hledat v čase O(log N).

Nejdelší rostoucí podposloupnost



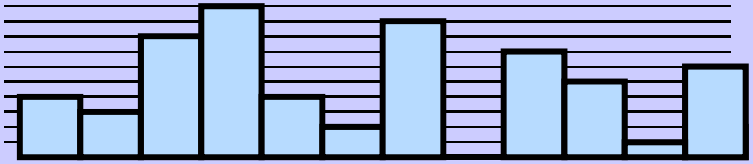
k .. index prvku
V[k] .. hodnota prvku
p[k] .. index předchůdce prvku
iL[d] .. index posledního prvku v rostoucí podposloupnosti délky d = 1, 2, ..., N.

Pro každý index k:
Nechť d je index max. prvku, pro který platí V[iL[d]] < V[k].
Potom iL[d+1] := k, p[k] = iL[d], pokud d existuje.
Jinak iL[1] := k, p[k] = null.

Pro každé k nalezneme d v čase O(log N) půlením intervalu.
Aktualizace iL a p proběhne v konstantním čase.
Celkem je složitost O(N log (N)).

Nejdelší rostoucí podposloupnost

k	1	2	3	4	5	6	7	8	9	10	11	12
V	4	3	8	10	4	2	9	0	7	5	1	6



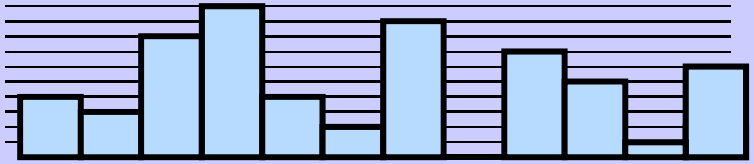
p	--											
iL	1											
V[iL]	4											

p	--	--										
iL	2											
V[iL]	3											

p	--	--	2									
iL	2	3										
V[iL]	3	8										

p	--	--	2	3								
iL	2	3	4									
V[iL]	3	8	10									

k	1	2	3	4	5	6	7	8	9	10	11	12
V	4	3	8	10	4	2	9	0	7	5	1	6



p	--	--	2	3	2							
iL	2	5	4									
V[iL]	3	4	10									

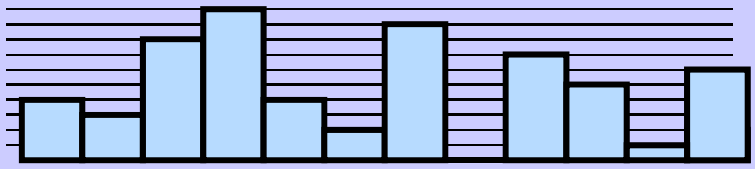
p	--	--	2	3	2	--						
iL	6	5	4									
V[iL]	2	4	10									

p	--	--	2	3	2	--	5					
iL	6	5	7									
V[iL]	2	4	9									

p	--	--	2	3	2	--	5	--				
iL	8	5	7									
V[iL]	0	4	9									

Nejdelší rostoucí podposloupnost

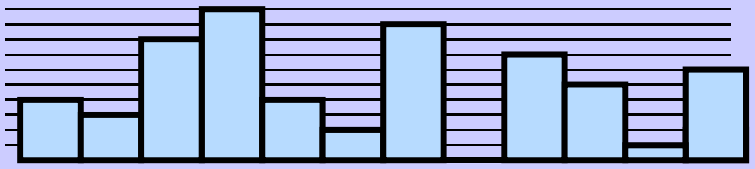
k	1	2	3	4	5	6	7	8	9	10	11	12
V	4	3	8	10	4	2	9	0	7	5	1	6



p	--	--	2	3	2	--	5	--	
iL	8	5	7						
V[iL]	0	4	9						

p	--	--	2	3	2	--	5	--	5
VL	8	5	9						
V[iL]	0	4	7						

k	1	2	3	4	5	6	7	8	9	10	11	12
V	4	3	8	10	4	2	9	0	7	5	1	6



p	--	--	2	3	2	--	5	--	5	5	
iL	8	5	10								
V[iL]	0	4	5								

p	--	--	2	3	2	--	5	--	5	5	8	
iL	8	11	10									
V[iL]	0	1	5									

p	--	--	2	3	2	--	5	--	5	5	8	10
iL	8	11	10	12								
V[iL]	0	1	5	6								

Rekonstrukce optimální cesty

Poslední definovaný prvek v iL je indexem posledního prvku jedné z optimálních podposloupností celé posloupnosti.
Pole p určuje pomocí předchůdců tuto podposloupnost.