Heuristic for “Go to Bucharest” that dominates SLD

- Array $A[i,j] = \text{straight-line distance (SLD) from city } i \text{ to city } j; \ B = \text{Bucharest}$;
- $s(n) = \text{successors of } n$;
- $c(m,n) = \{\text{if } (n \text{ in } s(m)) \text{ then } (\text{one-step road distance } m \text{ to } n) \text{ else } +\infty\}$;
- $s_k(n) = \text{all descendants of } n \text{ accessible from } n \text{ in exactly } k \text{ steps};$
- $S_k(n) = \text{all descendants of } n \text{ accessible from } n \text{ in } k \text{ steps or less};$
- $C_k(m,n) = \{\text{if } (n \text{ in } S_k(m)) \text{ then } (\text{shortest road distance } m \text{ to } n \text{ in } k \text{ steps or less}) \text{ else } +\infty\}$;
- $s, c, \text{ are computable in } O(b); s_k, S_k, C_k, \text{ are computable in } O(b^k)$.

- These heuristics both dominate SLD, and $h_2$ dominates $h_1$:
  - $h_1(n) = \min_{x \text{ in Romania}} (A[n,x] + A[x,B])$
  - $h_2(n) = \min_{x \text{ in } s(n)} (c(n,x) + A[x,B])$
- This family of heuristics all dominate SLD, and $i > j \Rightarrow h_i \text{ dominates } h_j$:
  - $h_k(n) = \min( (\min_{x \text{ in } S_k(n) \cap S_k(B)}) C_k(n,x) + C_k(x,B)),
  \ (\min_{x \text{ in } s_k(n), y \text{ in } s_k(B)} (C_k(n,x) + A[x,y] + C_k(y,B)))$
- $h_{\text{final}}(n) = \text{same as bidirectional search}; \Rightarrow \text{exponential cost}$