

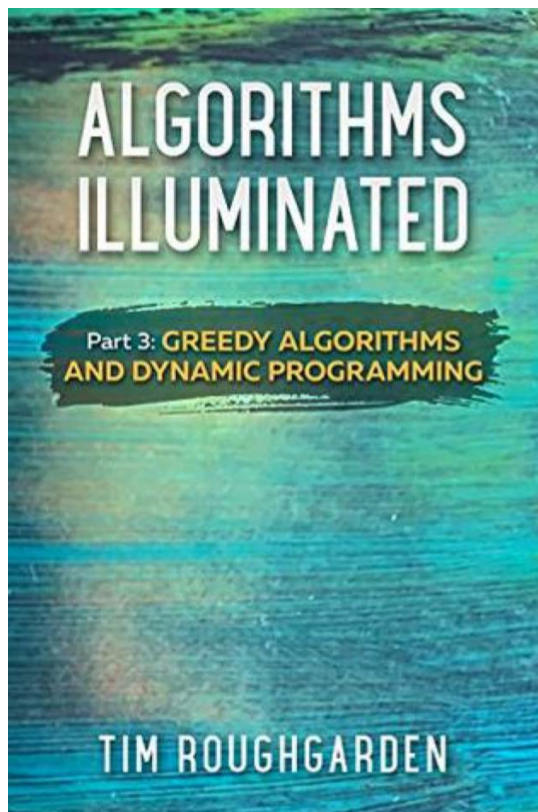
بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

طراحی الگوریتم (حریمانه)



دانشکده مهندسی برق و کامپیوتر، دانشگاه صنعتی اصفهان

بهار ۹۹

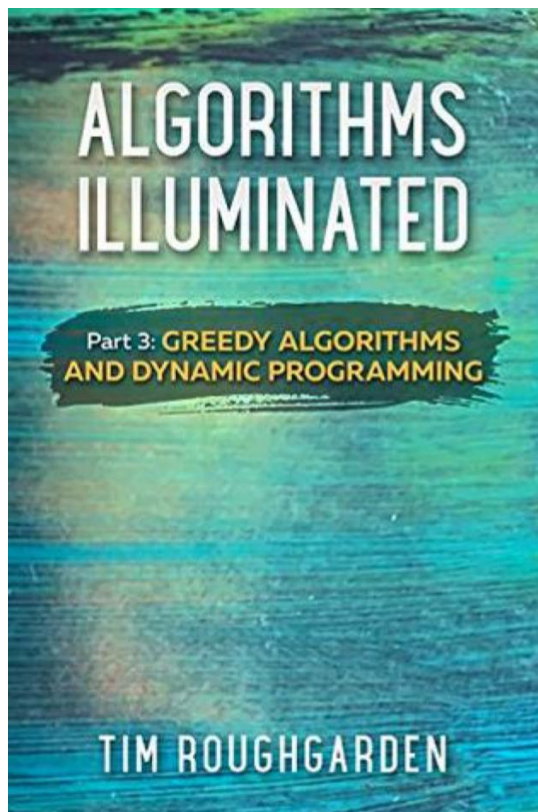


فصل چهاردهم، صفحه ۲۳



ورودی: برای هر سمبل یک فراوانی نامنفی.

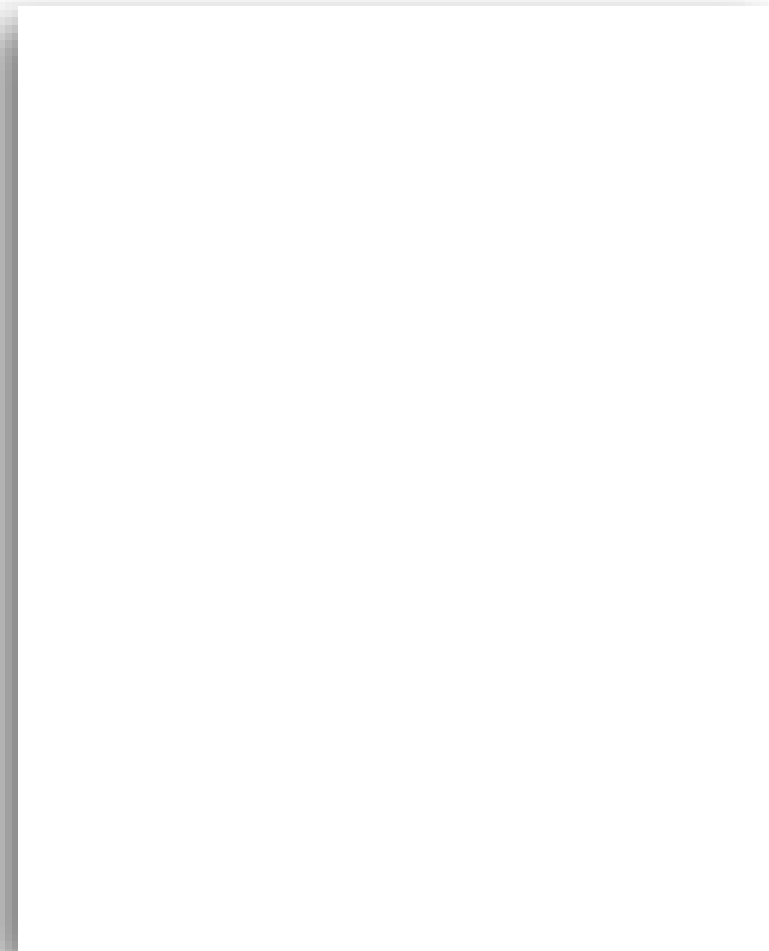
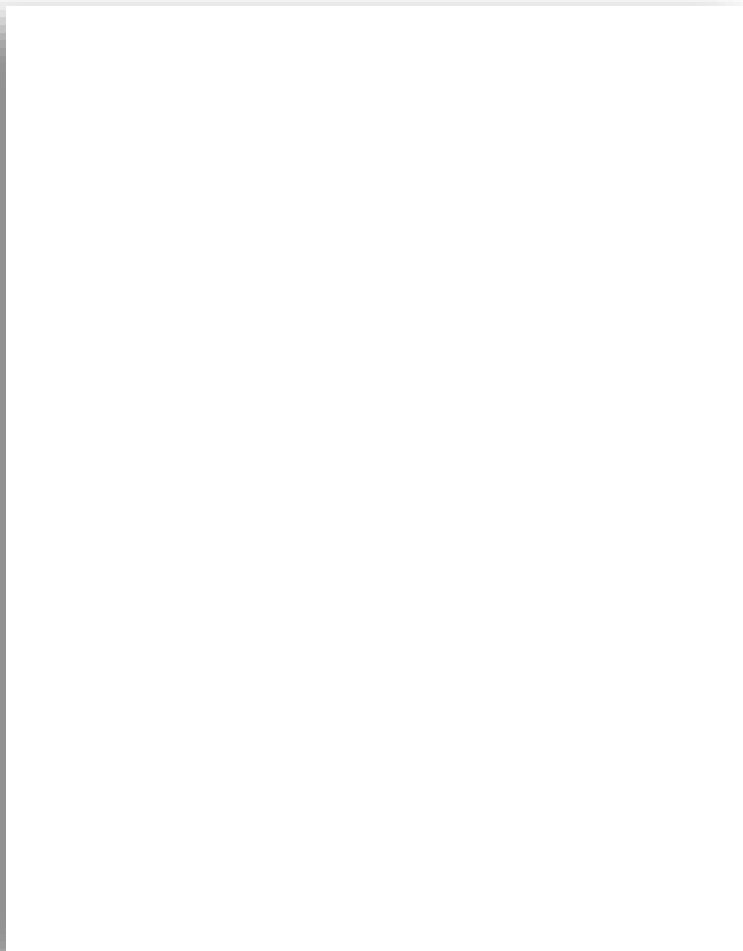
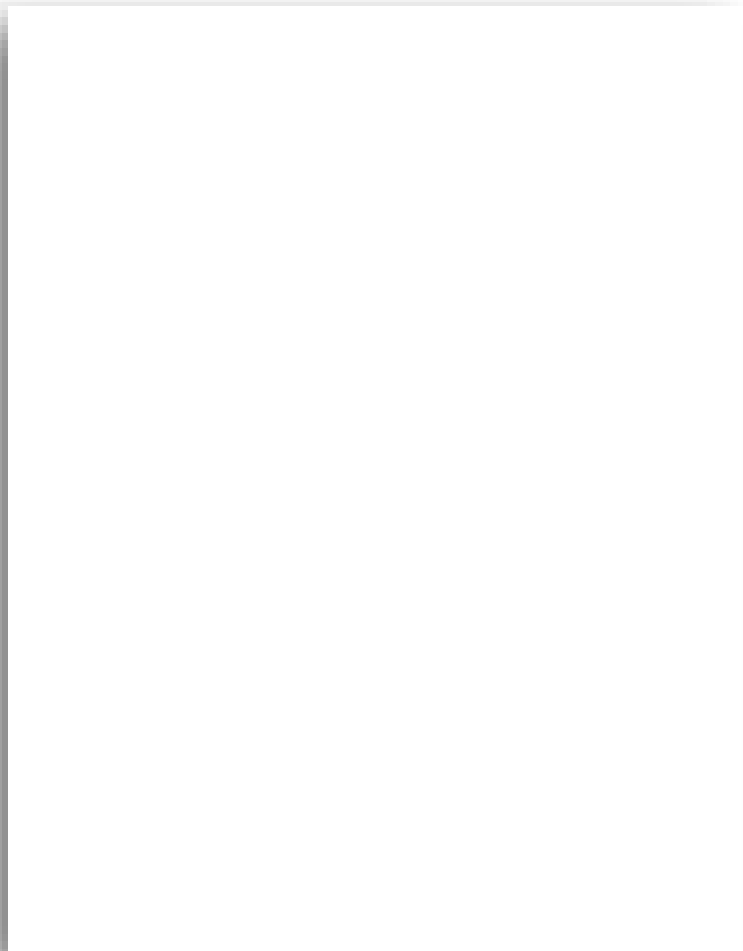
هدف: کد باینری بدون-پیشوند با کمترین میانگین طول کد گذاری.



فصل چهاردهم، صفحه ۲۳



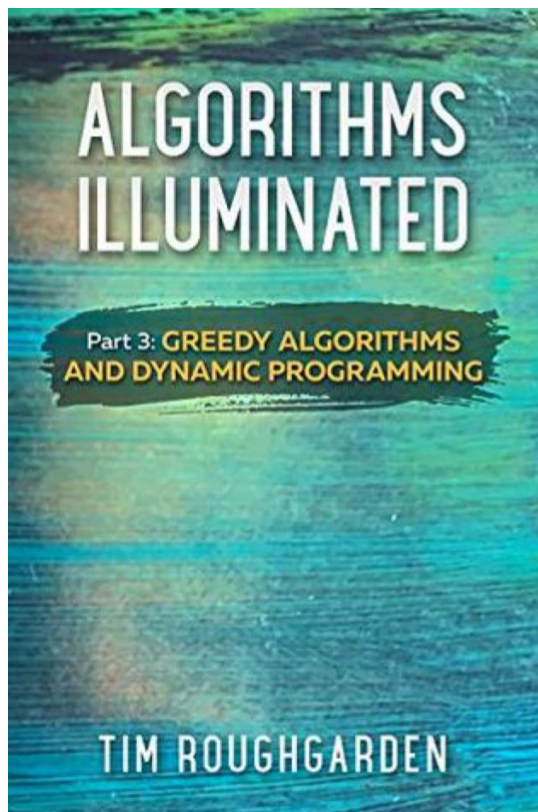
کدها به عنوان درخت





ورودی: برای هر سمبل یک فراوانی نامنفی.

هدف: یک درخت کدینگ با کمترین میانگین طول برگ‌ها.



فصل چهاردهم، صفحه ۲۳



ایده هافمن



Huffman

Input: a nonnegative frequency p_a for each symbol a of an alphabet Σ .

Output: the Σ -tree with minimum average leaf depth, representing the prefix-free binary code with minimum average encoding length.

// Initialization

for each $a \in \Sigma$ do

$T_a :=$ tree containing one node, labeled “ a ”

$P(T_a) := p_a$

$\mathcal{F} := \{T_a\}_{a \in \Sigma}$ // invariant: $\forall T \in \mathcal{F}, P(T) = \sum_{a \in T} p_a$

// Main loop

while \mathcal{F} contains at least two trees do

$T_1 := \operatorname{argmin}_{T \in \mathcal{F}} P(T)$ // min frequency sum

$T_2 := \operatorname{argmin}_{T \in \mathcal{F}, T \neq T_1} P(T)$ // second-smallest

 remove T_1 and T_2 from \mathcal{F}

 // roots of T_1, T_2 become left, right

 children of a new internal node

$T_3 :=$ merger of T_1 and T_2

$P(T_3) := P(T_1) + P(T_2)$ // maintains invariant

 add T_3 to \mathcal{F}

return the unique tree in \mathcal{F}



ایده اثبات درستی