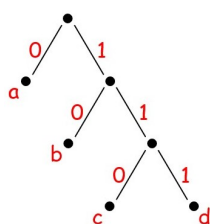




در جلسه‌ی قبل با مسئله‌ی کدگذاری آشنا شدیم. در این مسئله، الفبای  $C$  شامل  $n$  حرف را داریم و هدف ما اختصاص کد باینری به حروف آن است به گونه‌ای که هر حرف دقیقاً با یک کد معادل شود، هر متن کدشده به صورت یکتا بازیابی شود و متوسط طول متن تا حد امکان کم شود. برای این مسئله دو راه حل ارائه دادیم:

۱. کددهی با طول ثابت: در این روش به هر حرف یک کد با طول ثابت اختصاص می‌دهیم. کددهی به این روش یکتا است اما وقتی فرکانس حروف تفاوت زیادی با یکدیگر داشته باشند، این روش بهینه نیست.



حروف	a	b	c	d
کد اختصاص داده شده	۰	۱۰	۱۱۰	۱۱۱

۲. کددهی پیشوند آزاد: در این روش طول کدها متغیر بوده و کد هیچ حرفی پیشوند کد حرف دیگری نیست. یک نمونه از این روش کدگذاری به ازای الفبای  $\{a, b, c, d\}$  به شکل روبه‌رو است. هم‌چنین می‌توان آن را به شکل یک درخت نیز نشان داد.

## کدگذاری هافمن

این روش یک نمونه از روش کددهی پیشوند آزاد است. در این روش علاوه بر الفبا، فرکانس حروف آن نیز به عنوان ورودی گرفته می‌شود و خروجی یک روش کددهی به شکل یک درخت است که متوسط طول متن را کمینه می‌کند. الگوریتم به شرح زیر است:

۱. دو حرف با کمترین فرکانس را انتخاب و از الفبا حذف می‌کنیم.

۲. حرفی جدید با فرکانس برابر با جمع فرکانس دو حرف انتخاب شده می‌سازیم و به الفبا اضافه می‌کنیم.

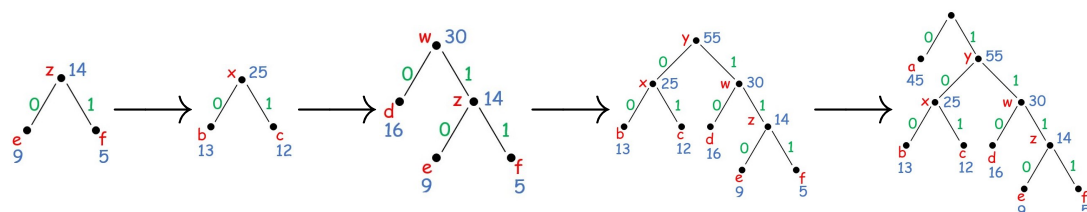
۳. دو حرف حذف شده را فرزندهای سمت راست و چپ حرف جدید قرار داده و مسئله را برای الفبای جدید به صورت بازگشتی

حل می‌کنیم تا زمانی که الفبا فقط یک حرف داشته باشد. آن حرف را ریشه درخت قرار می‌دهیم.

برای مثال فرض کنیم  $\{a, b, c, d, e, f\}$  حروف الفبا و فروانی آن‌ها مطابق جدول روبه‌رو باشد.

حروف	a	b	c	d	e	f
فراوانی	۴۵	۱۳	۱۲	۱۶	۹	۵

مراحل اجرای الگوریتم به شکل زیر است:



و کدهای خروجی الگوریتم مطابق جدول زیر است:

حروف	a	b	c	d	e	f
فراوانی	۰	۱۰۰	۱۰۱	۱۱۰	۱۱۱۰	۱۱۱۱

نشان دادیم که اگر  $x$  و  $y$  دو حرف با کمترین فرکانس باشند، آنگاه جواب بهینه‌ای وجود دارد که در آن این دو حرف تنها در بیت آخر متفاوت‌اند. در تمرین نشان خواهیم داد مقدار متوسط طول متن با این روش کمینه است.

با استفاده از داده‌ساختار هرم کمینه می‌توان این الگوریتم را در زمان  $\mathcal{O}(n \log n)$  پیاده‌سازی کرد. (چرا؟)

