Binomial Heap

هرم دو جمله‌ای

دانشجویان:

علی الیاسی، محمدعلی صابریان

هرم دو جمله‌ای

هیپ یا هرم، حالت خاصی از ساختمان داده درخت است که در آن کلید گره با فرزندانش مقایسه شده و بر همین اساس مرتب می‌شود.

براساس اینکه گره نسبت به فرزندانش چه وضعیتی دارد به دو دسته زیر تقسیم می‌شود:

Max-Heap

- کلید گره نسبت به فرزندانش بزرگتر یا مساوی است.

Min-Heap

- کلید گره نسبت به فرزندانش کوچکتر یا مساوی است.

انواع مختلف هرم

- Binary Heap

هرم باینری

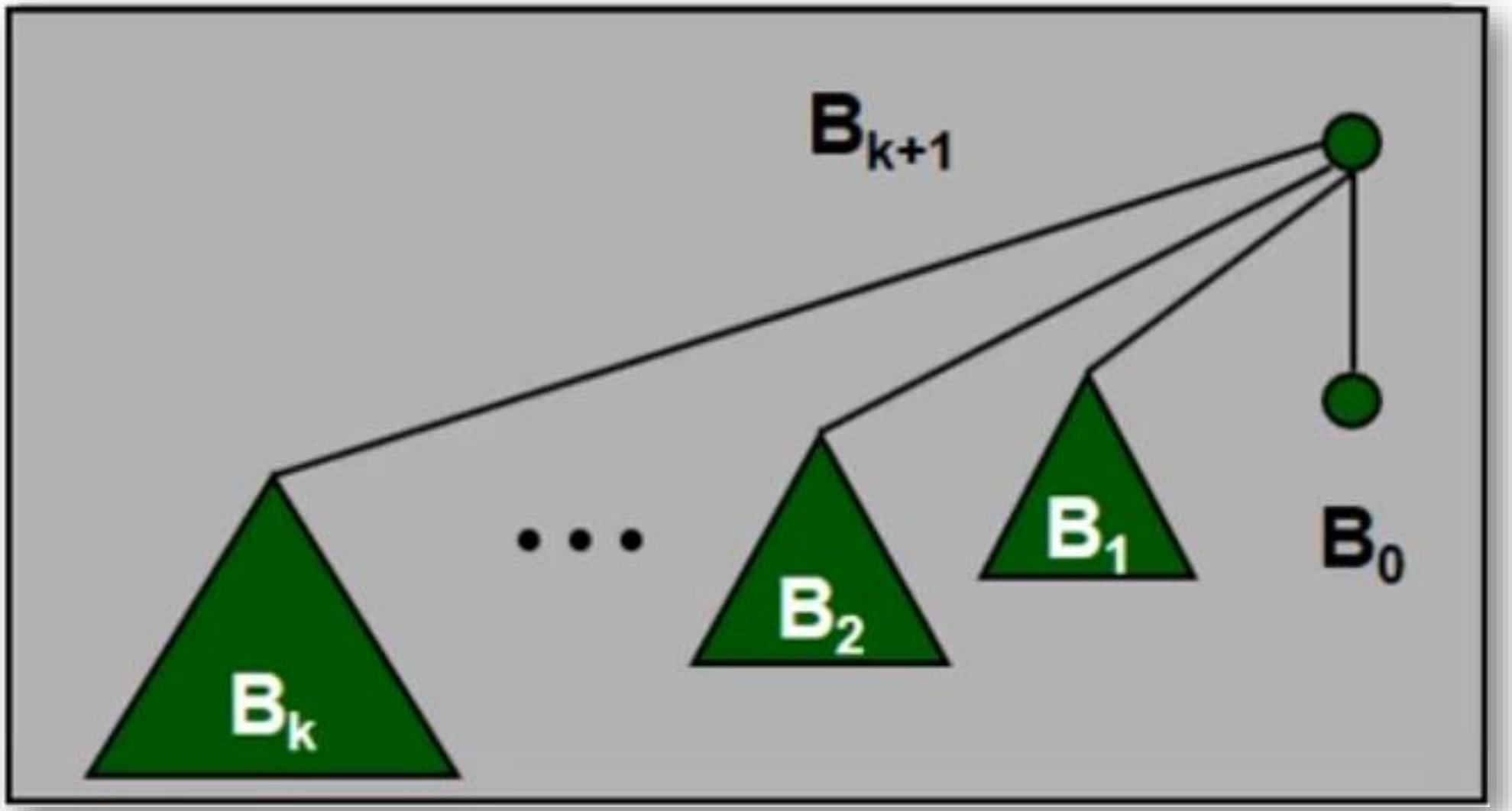
- Binomial Heap

هرم دو جمله‌ای

- Fibonacci Heap

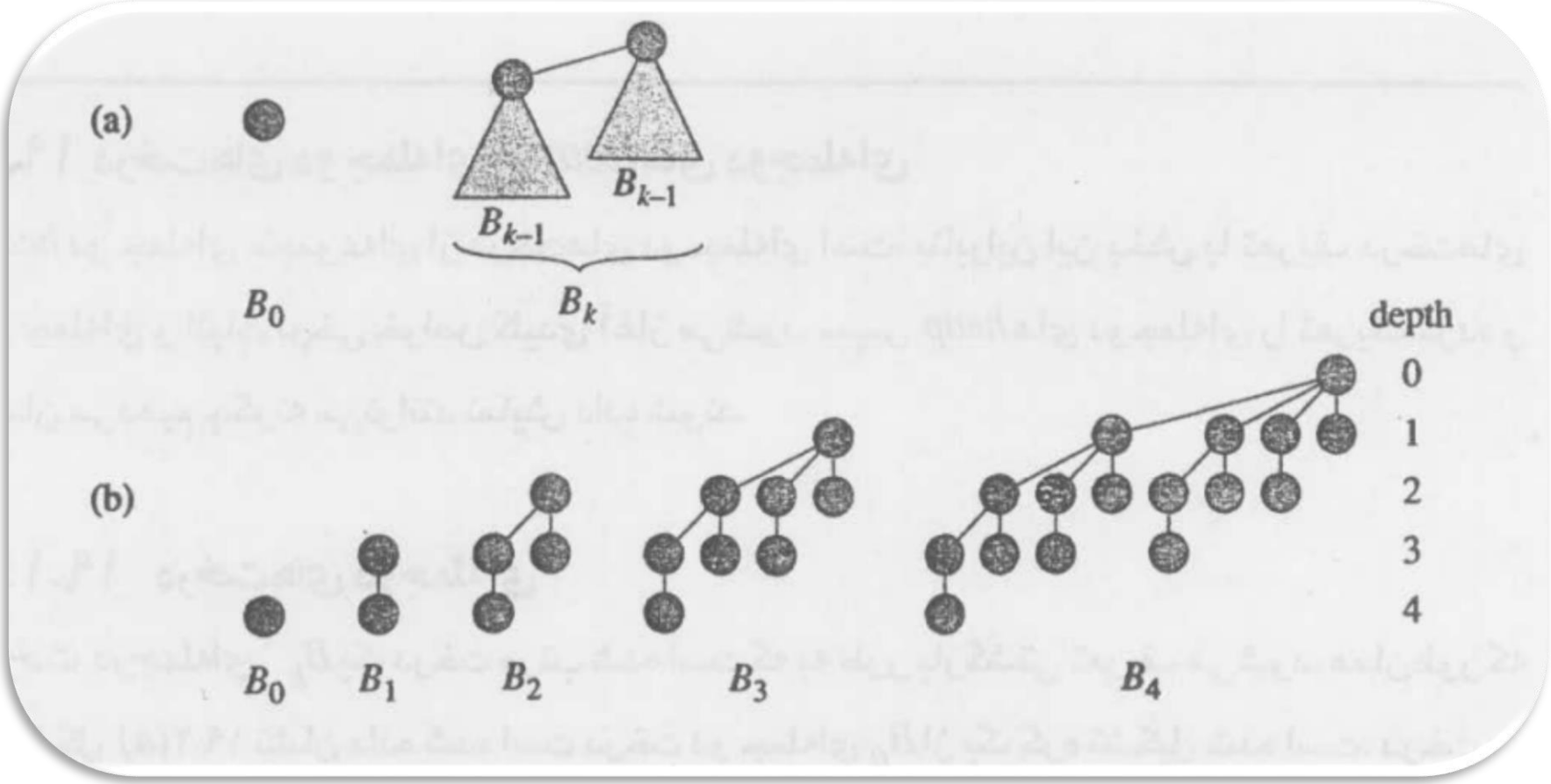
هرم فیبوناچی

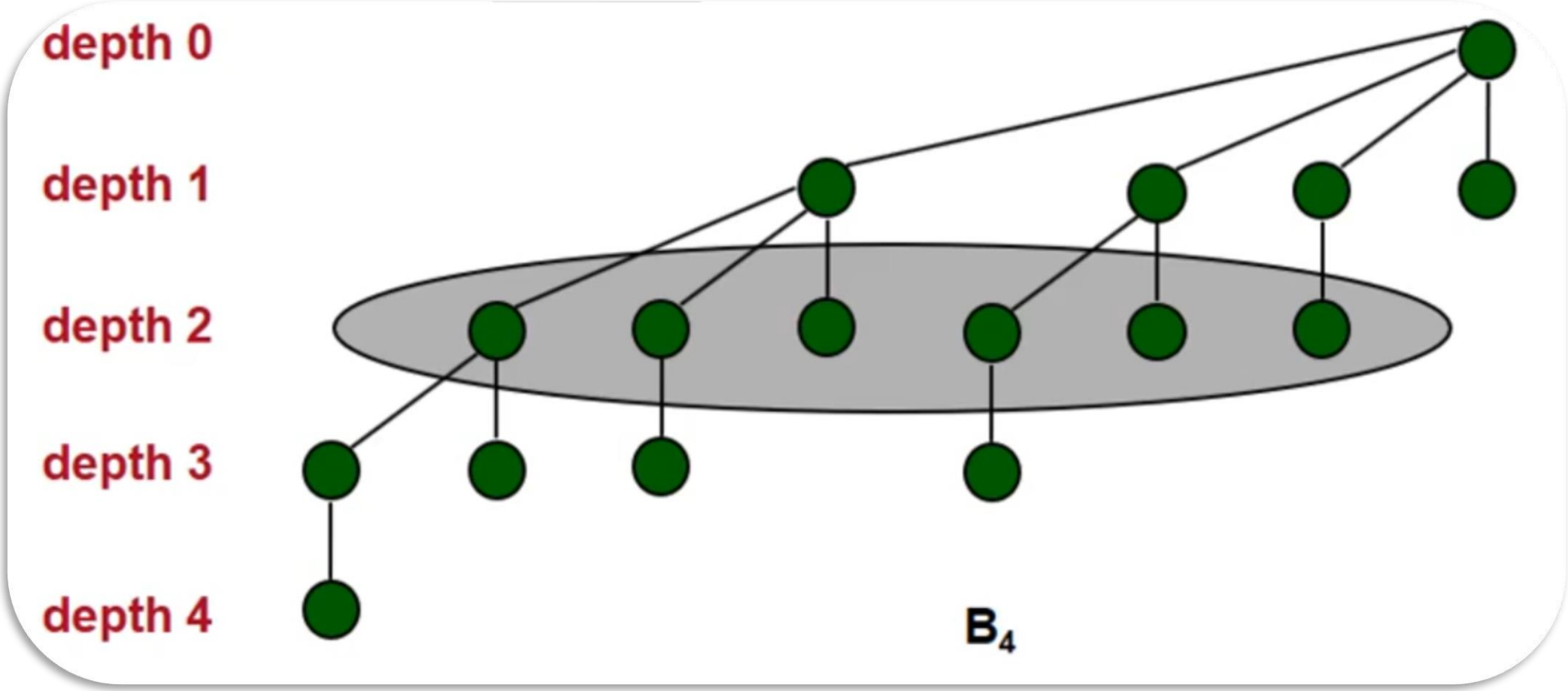
نمایش درخت دو جمله‌ای



درخت‌های دو جمله‌ای

درخت دو جمله‌ای B_k از دو درخت دو جمله‌ای B_{k-1} که به هم متصل شده‌اند تشکیل شده است: ریشه یکی، سمت چپ‌ترین فرزند ریشه دیگر است.




depth 0
depth 1
depth 2
depth 3
depth 4
B_4

ویژگی‌های درخت دوجمله‌ای B_k

- **در آن** 2^k گره وجود دارد.
- ارتفاع درخت برابر با k است.
- در عمق i، دقیقا $\binom{k}{i}$ گره وجود دارد.
- ریشه دارای درجه k است که از درجه‌های گره‌های دیگر بزرگتر است؛ بعلاوه اگر فرزندان ریشه از چپ به راست با $0,1,\ldots$ شماره‌گذاری شوند ، آنگاه i ریشه زیر درخت B_i است.

هرم‌های دو جمله‌ای

هرم دو جمله‌ای، مجموعه‌ای از درخت‌های دو جمله‌ای است که ویژگی‌های هرم دو جمله‌ای که در زیر ذکر شده‌اند را داراست:

۱. هر درخت دو جمله‌ای از ویژگی min-heap پیروی می‌کند: کلید یک گره، بزرگتر یا مساوی کلید پدرش است. می گوییم چنین درختی min-heap مرتب شده است.

۲. برای عدد صحیح غیر منفی k، حداکثر یک درخت دو جمله ای که ریشه‌ای با درجه k دارد وجود دارد.

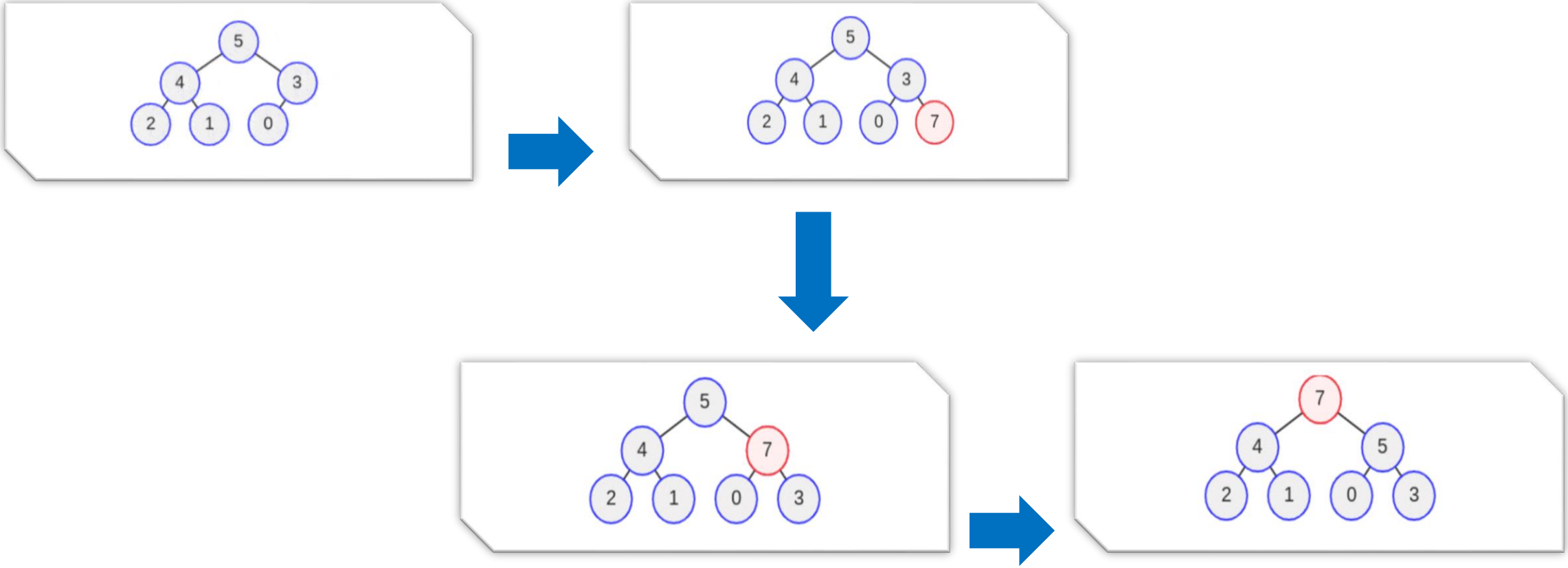
نکته: در هرم دو جمله‌ای حداکثر $\lfloor logn \rfloor + 1$ درخت دو جمله ای وجود دارد.

زمان اجرای اعمال مختلف بر روی هرم‌های مختلف

Procedure	Binary heap (worst-case)	Binomial heap (worst-case)	Fibonacci heap (amortized)
MAKE-HEAP	$\Theta(1)$	$\Theta(1)$	$\Theta(1)$
INSERT	$\Theta(\lg n)$	$O(\lg n)$	$\Theta(1)$
MINIMUM	$\Theta(1)$	$O(\lg n)$	$\Theta(1)$
EXTRACT-MIN	$\Theta(\lg n)$	$\Theta(\lg n)$	$O(\lg n)$
UNION	$\Theta(n)$	$O(\lg n)$	$\Theta(1)$
DECREASE-KEY	$\Theta(\lg n)$	$\Theta(\lg n)$	$\Theta(1)$
DELETE	$\Theta(\lg n)$	$\Theta(\lg n)$	$O(\lg n)$

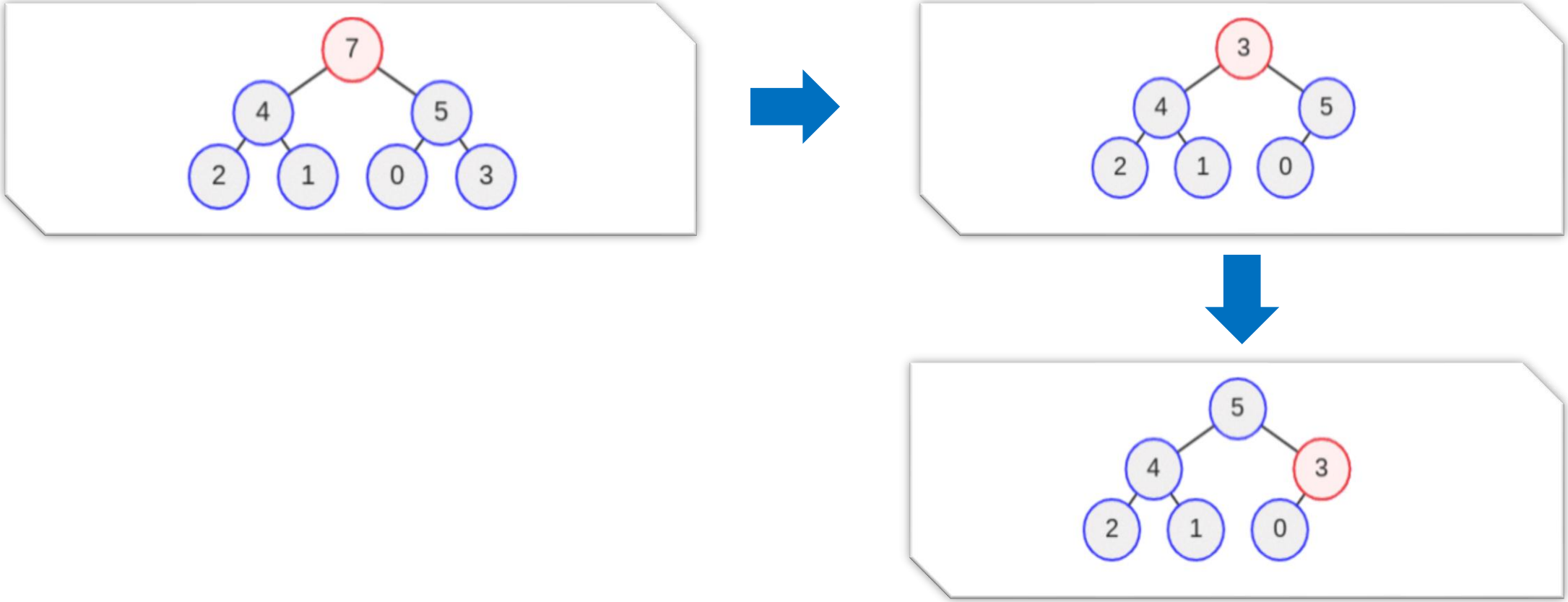
شکل ۱۹.۱ زمان اجرا برای اعمال بر روی سه پیاده سازی *heap*های قابل ادغام. تعداد اقلام در *heap* (ها) در زمان یک عمل به وسیله n نشان داده می‌شود.

درج یا Insert



حذف یا Delete

در عمل حذف همواره مقدار ریشه حذف شده و سمت راست‌ترین عنصر موجود در پایین‌ترین سطح، در ریشه قـرار می گیرد و درخت مجدداً تنظیم می شود.



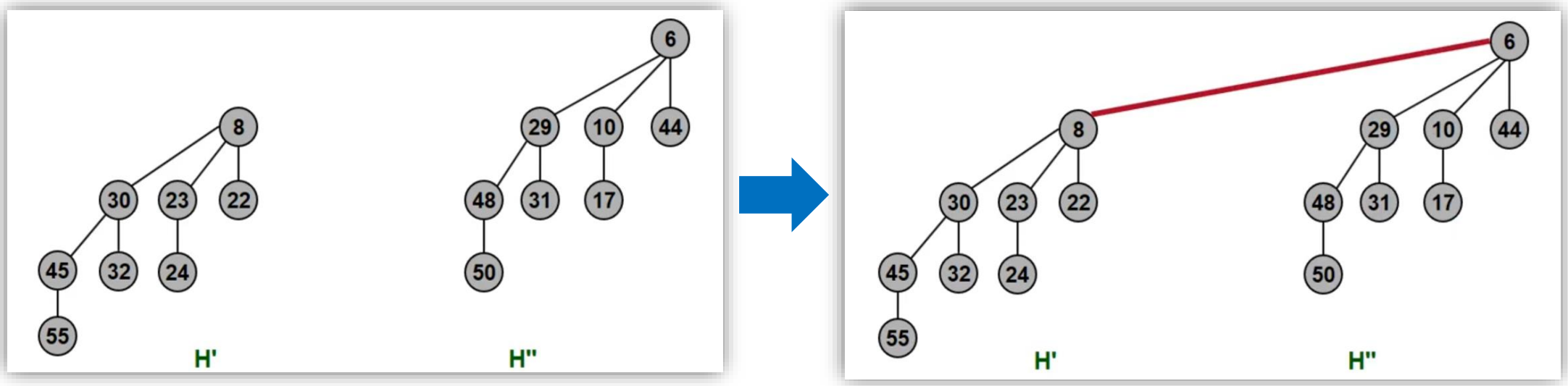
اجتماع در هرم ها یا Union

این عملیات با دقت و قابلیت اطمینان انجام می‌شود تا پس از ادغام، هرم حاصل از عملیات ادغام تمامی خصوصیات و خواص هر دو هرم اصلی را حفظ کند. از این عملیات در الگوریتم ها و سیستم هایی که از هرم به عنوان یکی از ساختارهای داده اصلی استفاده می‌کنند، به منظور ادغام داده‌ها و بهبود عملکرد الگوریتم‌ها استفاده می‌شود.

اجتماع در هرم‌ها – دو درخت هم‌اندازه

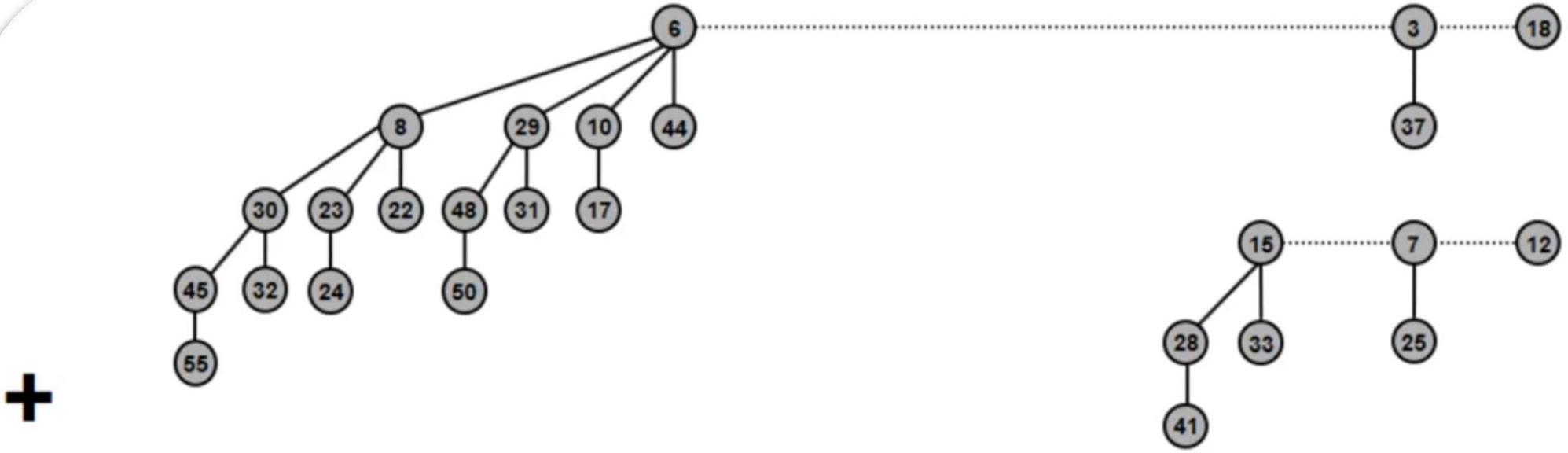
۱. دو ریشه درخت را به هم متصل می کنیم.

۲. درخت دو جمله ای با ریشه کوچک‌تر را در سطح پایین‌تر قرار می‌دهیم (Min-heap)



Max-heap

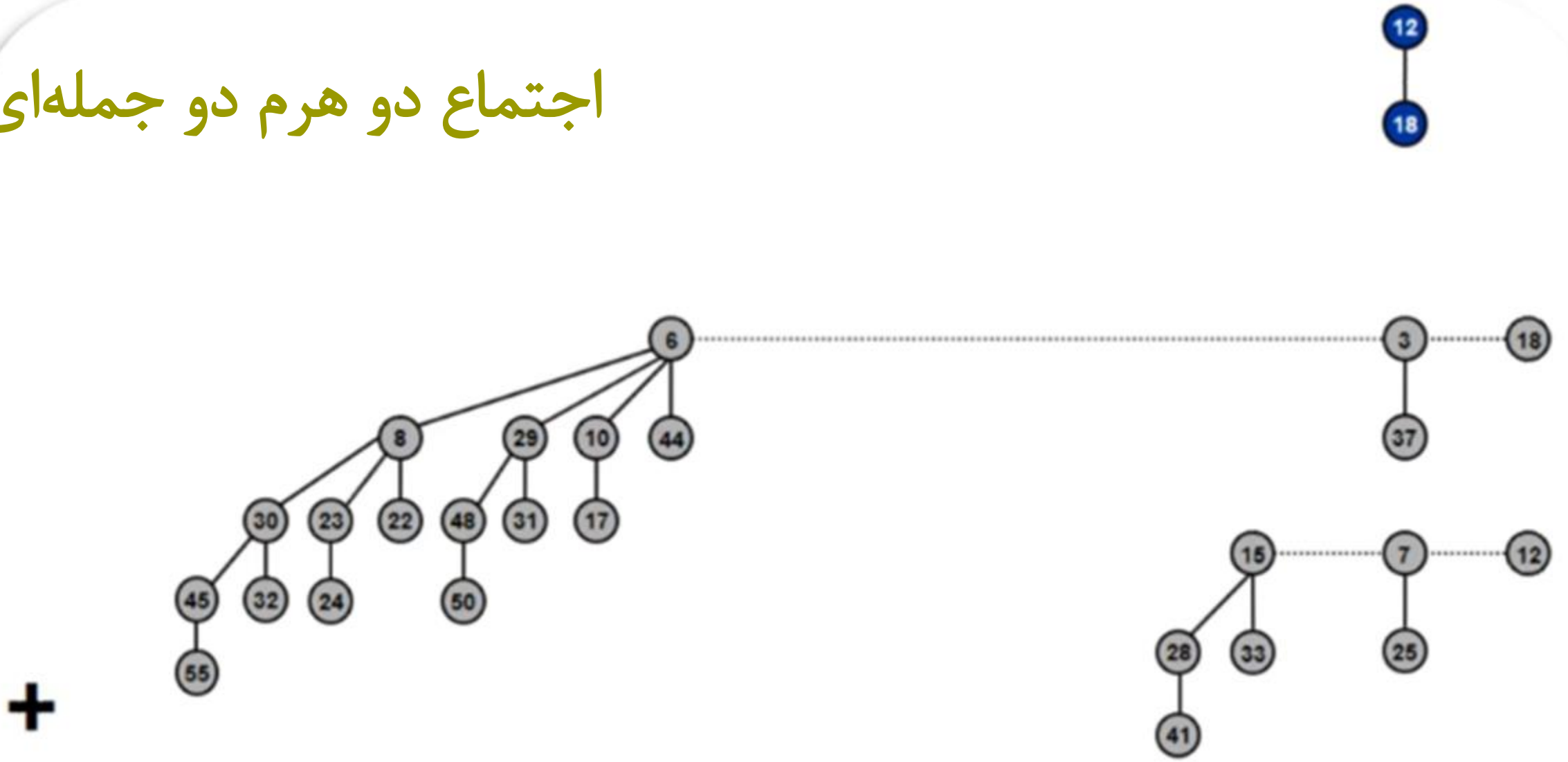
اجتماع دو هرم دو جمله‌ای



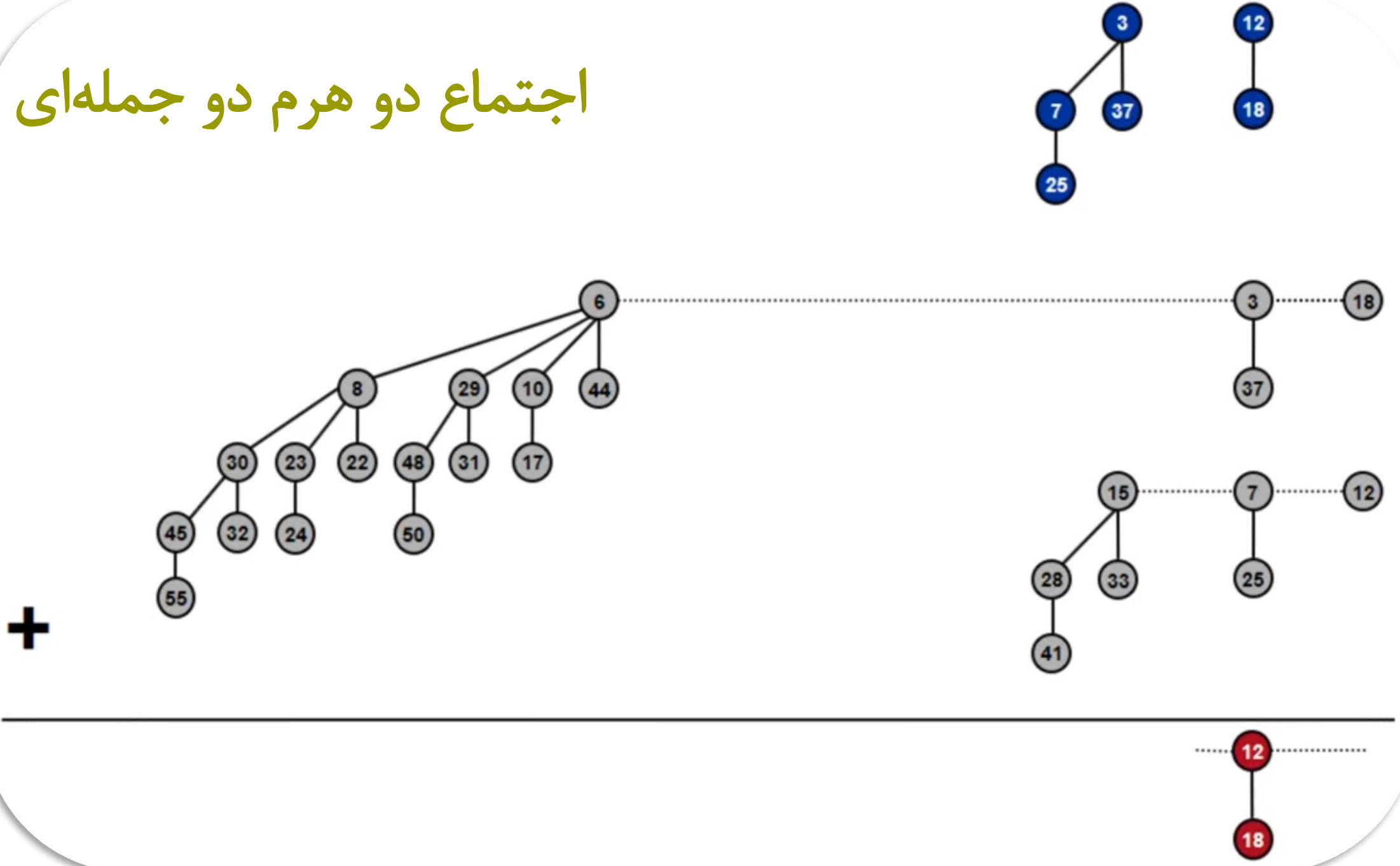
19 + 7 = 26

		1	1	1	
	1	0	0	1	1
+	0	0	1	1	1
	1	1	0	1	0

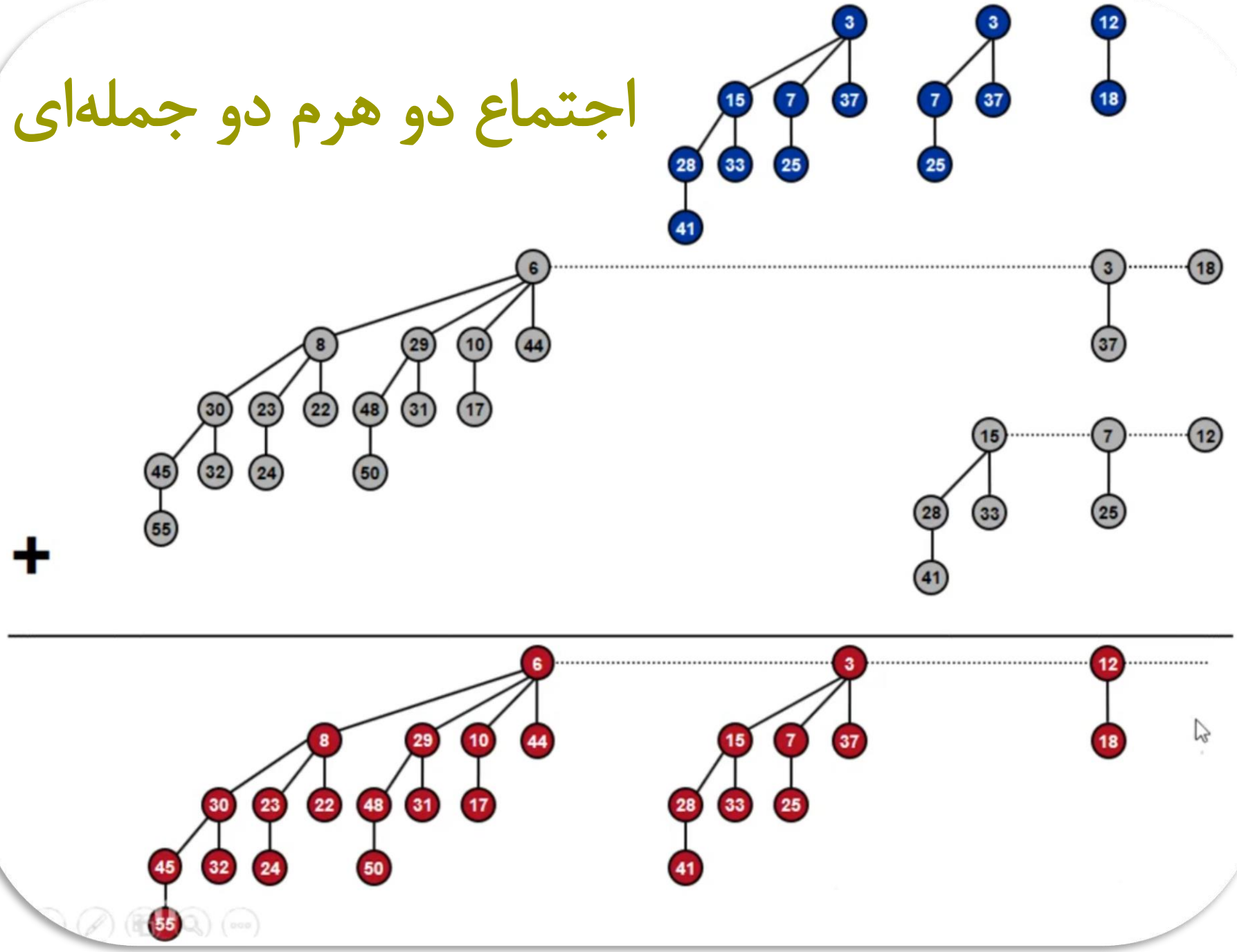
اجتماع دو هرم دو جمله‌ای



اجتماع دو هرم دو جمله‌ای



اجتماع دو هرم دو جمله‌ای



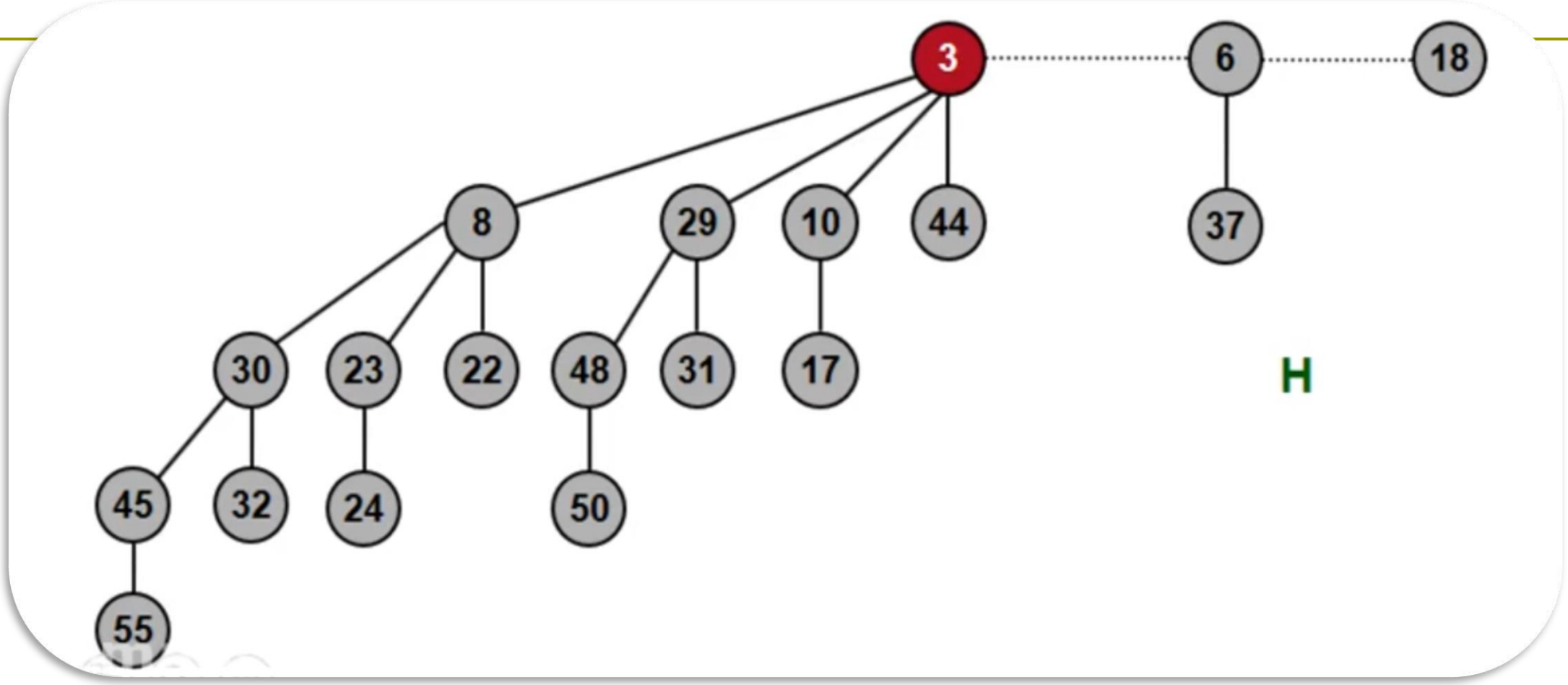
حذف کمینه در هرم

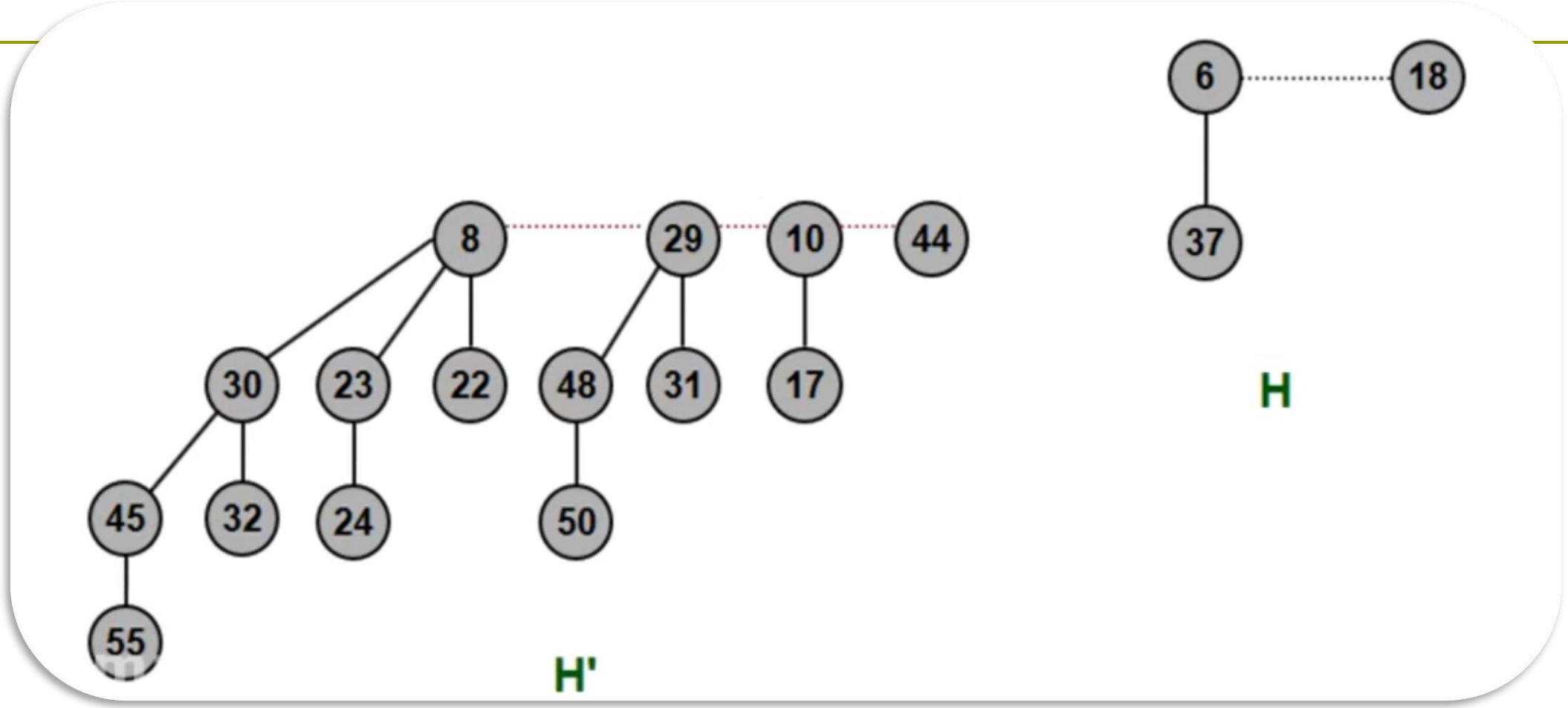
۱. ابتدا در بین ریشه‌ها با جستجوی خطی عنصر کمینه را پیدا کرده و آن را حذف می‌کنیم.

۲. با حذف گره، مجموعه‌ای از درخت‌های دو جمله‌ای خواهیم داشت.

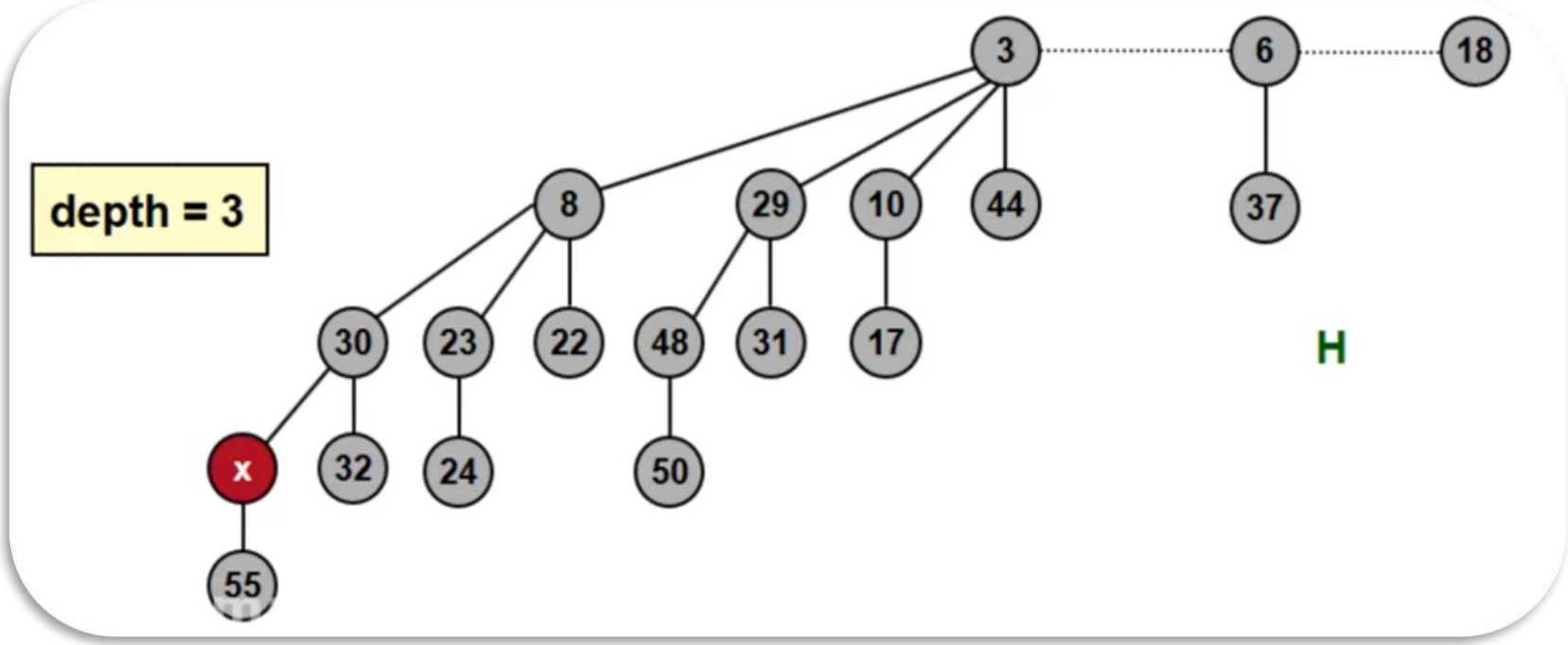
۳. مجموعه درخت‌های دو جمله‌ای باقیمانده فرزند را یک هرم دو جمله‌ای در نظر می‌گیریم.

۴. هرم بدست آمده را با مابقی هرم قبلی ادغام می‌کنیم.


3
6
18
8
29
10
44
37
30
23
22
48
31
17
H
45
32
24
50
55


8
29
10
44
30
23
22
48
31
17
45
32
24
50
55
H'
6
18
37
H

Decrease Key در هرم های دو جمله ای:



حذف در هرم‌های دو جمله‌ای یا Delete

1. ابتدا عنصر مورد نظر را به $-\infty$ تغییر می‌دهیم و مرتب می‌کنیم.
2. عنصر $-\infty$ از همه عناصر کوچکتر بوده پس در ریشه قرار می‌گیرد.
۳. سپس با استفاده از حذف ریشه آن را حذف می‌کنیم.