به نام خدا

|  |  |  |  |
| --- | --- | --- | --- |
| مدرس: مظفر بگ محمدي | دانشگاه ايلام | ترم اول سال تحصيلي 01-00 | سري پنجم تمرينات ساختمان داده |

1. اعداد 2, 25, 4, 8, 12, 7, 63, 10, 9, 14را به ترتیب وارد یک درخت جستجوی دودویی کنید سپس ریشه را حذف کنید.
2. درخت معادل عبارت زیر را رسم کنید. سپس معادل پسوندی آنرا با پیمایش مناسب درخت بدست آورید.

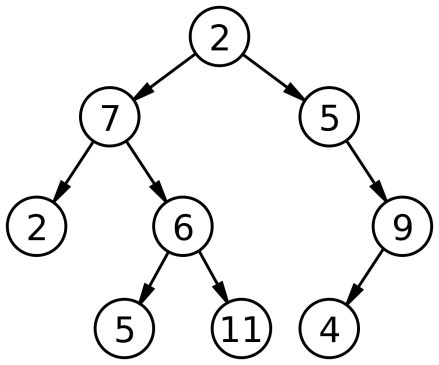
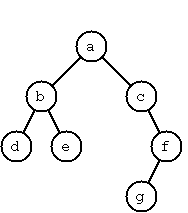
(1 - 2) / 3 + (4 \* 5) - (6 % 7)

1. با استقراء ریاضی نشان دهید که یک درخت دودویی که N نود داشته باشد، دقیقا دارای N+1 لینک خالی است.
2. heap حداقل دودویی معادل با آرایه A = 3, 14, 17, 2, 11, 43, 37, 6, 1, 19 را بدست آورید. وضعیتهای heap را بعد از اضافه کردن هر عدد رسم کنید. نمایش آرایه ای heap را بدست آورید. سپس، وضعیت توده را بعد از حذف کوچکترین عنصر نشان دهید.
3. پیمایش preorder و inorder یک درخت دودویی داده شده اند، درخت را بدست آورید و پیمایش postorder آنرا نمایش دهید.

Preorder: 4, 2, 5, 1, 3, 6

Inorder: 1, 2, 4, 5, 3, 6

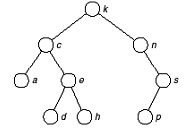
1. پيمايش post-order يک درخت جستجوي دودويي به صورت 1, 7, 5, 50, 40, 10 است. درخت را بدست آوريد.
2. يک درخت دودويي داريم که در هر نود آن يک حرف ذخيره شده است. پيمايش preorder درخت برابر ANUPSIDEDOWNTREE و پيمايش inorder آن برابر UNPADIESDNWTOERE است. درخت را رسم کنيد.
3. پیمایشهای مختلف درختان زیر را بدست آورید.

1. درخت مربوط به آرایه زیر را رسم کنید.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 12 | 7 | 19 | 2 | 4 | 5 | 32 |

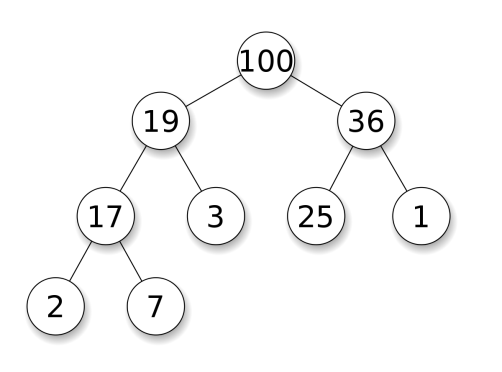
1. درخت جستجوی دودویی زیر را در نظر بگیرید. عناصر q و r را به درخت اضافه کنید و درخت را رسم کنید.



1. عنصر 12 را به heap حداکثر زیر اضافه کنيد. سپس همه عناصر را حذف کنيد. تمام مراحل مياني را نشان دهيد.

Max Heap in Java - GeeksforGeeks

1. نمایش آرایه‌ای درخت را زیر را مشخص کنید.



1. تمام پیمایشهای درخت فوق را مشخص کنید.
2. اگر یک درخت دودویی کامل شامل n نود باشد، حداقل تعداد و حداکثر تعداد سطوح درخت بر حسب n چقدر است؟
3. اعداد 3 8 1 7 0 6 9 4 11 5 2 را به ترتیب از راست به چپ وارد یک درخت جستجویی دودویی کنید. حال نود ۸ را حذف کنید.
4. قسمتی از پیمایشهای یک درخت جستجوی دودویی به صورت زیر است. قسمتهای نامعلوم با x مشخص شده‌اند. درخت را رسم کنید.

|  |  |
| --- | --- |
| x, x, x, x, 7, 12 | پیش ترتیب |
| x, 2, x, 5, x, 10 | پس ترتیب |
| x, 4, x, x, x, x | به ترتیب |

1. تابعی بنویسید که تعداد نودهای غیر انتهایی یک درخت دودویی را محاسبه کند.
2. یک متد بنویسید که تعداد نودهای برگ یک درخت دودویی را بشمارد.
3. یک متد بنویسید که تمام نودهایی که مقدار آنها از a بزرگتر هستند را حذف کند.
4. متدی بنویسید که ارتفاع درخت را محاسبه کند.
5. درخت جستجوی دودویی را برای اعداد طبیعی پیاده کنید. سپس در یک حلقه ۵۰ عدد تصادفی بین ۱ تا ۱۰۰ را وارد درخت کنید. پیمایش به ترتیب درخت را نمایش دهید. سپس چند نود را به صورت تصادفی حذف کنید و پیمایش به ترتیب درخت را دوباره نمایش دهید.
6. کدی بنویسید که پیمایشهای به ترتیب و پس ترتیب را دریافت کند و پیمایش پیش ترتیب را چاپ کند.
7. کدی بنویسید که یک آرایه تصادفی از اعداد طبیعی را دریافت کند. سپس، آرایه را به توده حداکثر تبدیل کنید. حال اعمال زیر را روی توده از چپ به راست انجام دهید.

add(4), add(18), add(11), add(2), delete(), delete(), add(13), add(5), add(1), delete()