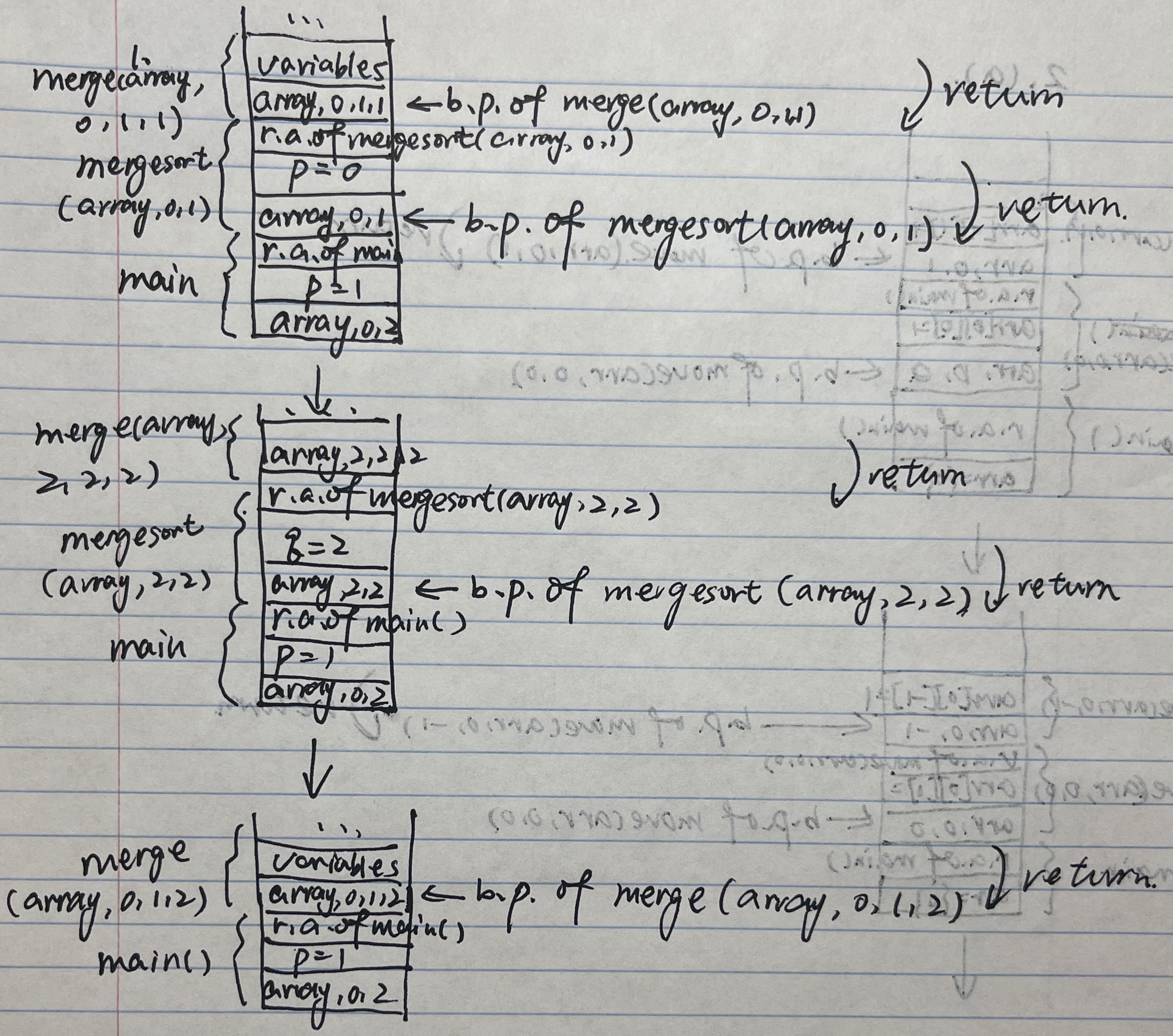
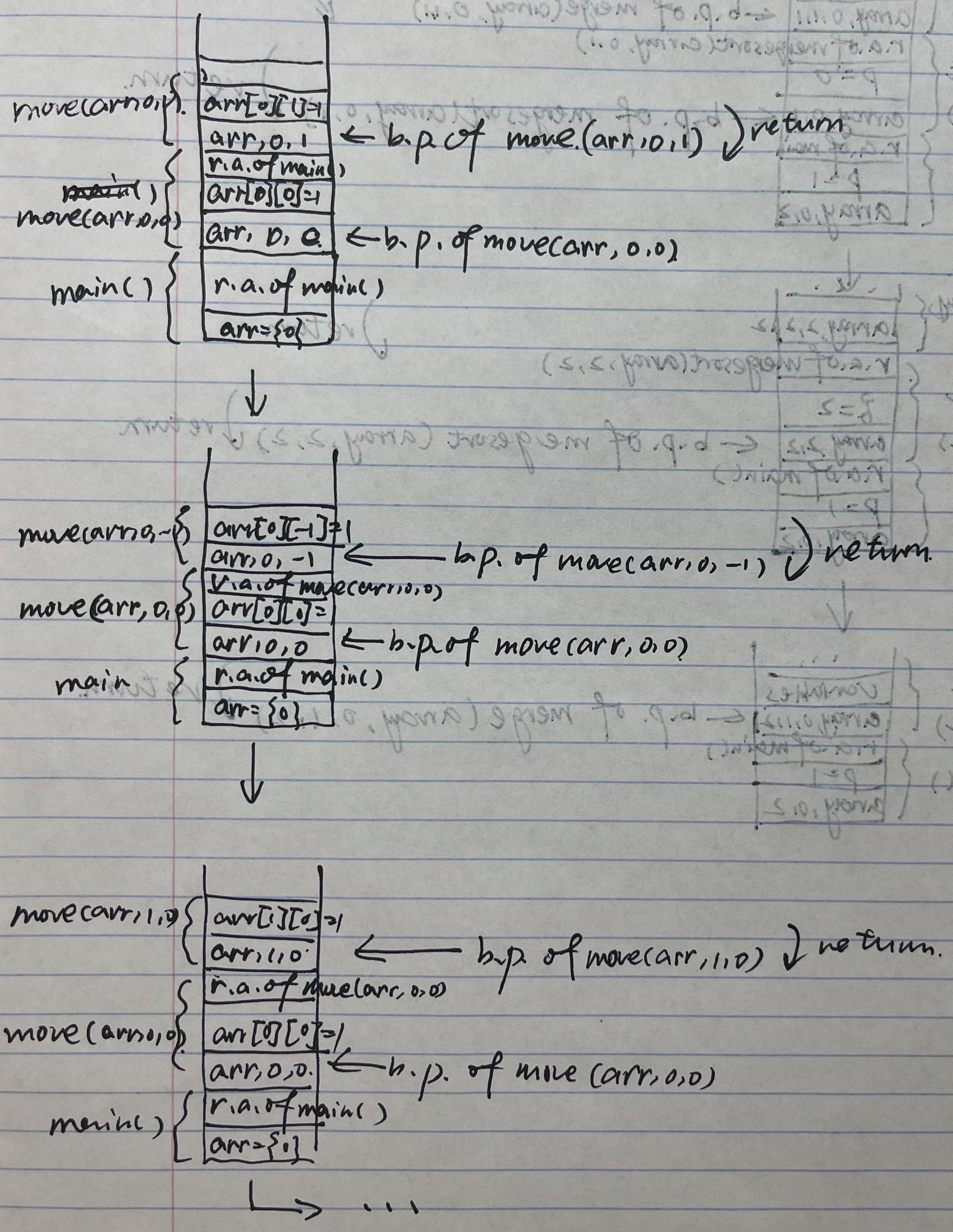

...
merge(array, 0, 1, 1)
variables
array, 0, 1, 1
← b.p. of merge(array, 0, 1, 1)
return
mergesort (array, 0, 1)
r.a. of mergesort(array, 0, 1)
p = 0
array, 0, 1
← b.p. of mergesort(array, 0, 1)
return.
main
r.a. of main
p = 1
array, 0, 2
...
merge(array, 2, 2, 2)
array, 2, 2, 2
return
mergesort (array, 2, 2)
r.a. of mergesort(array, 2, 2)
q = 2
array, 2, 2
← b.p. of mergesort (array, 2, 2)
return
main
r.a. of main()
p = 1
array, 0, 2
...
merge (array, 0, 1, 2)
variables
array, 0, 1, 2
← b.p. of merge (array, 0, 1, 2)
return.
main()
r.a. of main()
p = 1
array, 0, 2

2. (a).



move(arr,0,1)
arr[0][1]=1
arr, 0, 1
← b.p of move(arr,0,1) return.
~~main~~)
move(arr,0,0)
r.a. of main()
arr[0][0]=1
arr, 0, 0
← b.p. of move(arr, 0,0)
main()
r.a. of main()
arr={0}
move(arr,0,-1)
arr[0][-1]=1
arr, 0, -1
← b.p. of move(arr, 0, -1) return.
move(arr, 0,0)
r.a. of move(arr,0,0)
arr[0][0]=
arr, 0, 0
← b.p. of move(arr, 0,0)
main
r.a. of main()
arr={0}
move(arr,1,0)
arr[1][0]=1
arr, 1, 0
← b.p. of move(arr,1,0) return.
move(arr,0,0)
r.a. of move(arr,0,0)
arr[0][0]=1
arr, 0, 0.
← b.p. of move (arr, 0,0)
main()
r.a. of main()
arr={0}
...

6. The program assigns values to the memory that's not allocated for "arr".

3. (a). let $n_0=1, c_1=\frac{1}{2}$.

Then $\frac{1}{2}n^2-3n \le \frac{1}{2}n^2$ is correct for $\forall n \ge 1$.

So $\frac{1}{2}n^2-3n = O(n^2)$

let $n_0=12, c_2=\frac{1}{4}$

Then $\frac{1}{2}n^2-3n \ge \frac{1}{4}n^2 \Leftrightarrow 2n^2-12n \ge n^2$

$\Leftrightarrow n^2 \ge 12n \Leftrightarrow n \ge 12 = n_0$

So $\frac{1}{2}n^2-3n = \Omega(n^2)$

So $\frac{1}{2}n^2-3n = \Theta(n^2)$

(b). let $f(n)=n^3, g(n)=n^2$

$\lim\limits_{n\to\infty}\frac{g(n)}{f(n)} = \frac{1}{n} = 0$, So $f(n)=\omega(g(n))$

So $n^3 \ne O(n^2)$

(c). let $f(n)=n\log n-2n+13$, $g(n)=n\log n$

$\lim\limits_{n\to\infty}\frac{f(n)}{g(n)} = \frac{n\log n-2n+13}{n\log n} = \lim\limits_{n\to\infty}\left(1-\frac{2}{\log n}+\frac{13}{n\log n}\right) = 1$

So $n\log n-2n+13 = \Theta(n\log n)$

So $n\log n-2n+13 = \Omega(n\log n)$

(d). let $f(n)=n^{1/2}$, $g(n)=n^{2/3}$

Since $\lim_{n\to\infty} \frac{f(n)}{g(n)} = \lim_{n\to\infty} \frac{n^{1/2}}{n^{2/3}} = \lim_{n\to\infty} n^{-\frac{1}{6}} = 0$

So $f(n) \neq \Theta(g(n))$

(e). ~~let f(n)=n!~~

let $c=1$, $n_0=4$,

then we have $n! > 2^n$ is correct when $n=4$

Assume $n! > 2^n$ is correct for $n=k \geq 4$

then $(k+1)! = (k+1)\cdot k! > (k+1)\cdot 2^k \geq 5\cdot 2^k > 2^{k+1}$

$n! > 2^n$ holds true for $k+1$

So. $n! > 2^n$ is correct for $n \geq 4$.

So. $n! = \omega(2^n)$

4. $T(n) = 1 \times T(\frac{n}{2}) + \frac{n}{2}$

let $f(n) = \frac{n}{2}$, then $f(n) = \Theta(n)$

since. $\log_b a = \log_2 1 = 0 < 1$

So $T(n) = O(f(n)) = O(n)$