

سؤال 1:

ا) نادرست، زیرا در عملیات insert برای آرایه ها در صورت حذف و یا اضافه کردن یک داده لازم است سایر داده ها به اندازه یک خانه به عقب یا جلو شیفیت پیدا کنند.

ب) نادرست، در این عملیات بدلیل دفعه بودن آرایه تفاوت چندانی با لینک لیست احساس نمی شود البته اگر لینک لیست از نوع ساختار داده دوفیلد بودن باشد، به مراتب عملیات سریع در آن آسانتر است. (در مرتبه $\log_2 n$) اما در غیر این صورت می توان گفت عملیات سریع در آرایه دفعه آسانتر است، زیرا داده های یک آرایه در حافظه پشت سر هم هستند و به نظر میرسد که بررسی داده های پشت سر هم برای کامپیوتر آسانتر است.

ج) درست، البته باز بستگی به نوع لینک لیست دارد. اما با فرض اینکه لیست پیوندی یک طرفه باشد می توان گفت که عملیات سریع در یک آرایه مرتب شده به مراتب آسانتر است. (در مرتبه $\log_2 n$) اما برای لینک لیست تعداد چک کردن از مرتبه n می باشد.

د) نادرست، زیرا در عملیات merge در لیست پیوندی به راحتی می توان $node \rightarrow next$ را از یک لیست را برابر با آدرس اولین $node$ لیست دیگر قرار داد که به راحتی انجام می شود. چون فقط یک آدرس را تغییر می دهیم. اما در آرایه ها نیاز به $realloc$ داریم که به نسبت کار سنگین و وقتگیرتری برای کامپیوتر است.

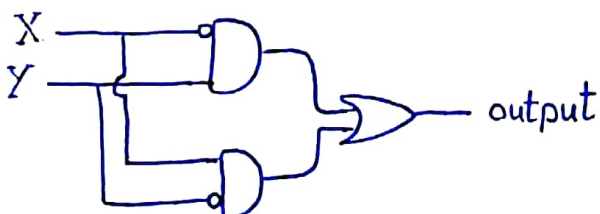
سؤال 2:

به طور کلی تابع f بر هر n node ها را چاپ می کند بعضی جاها دوباره $node$ ها را از خود لیست اضافه می کند. اما متغیری که در نهایت برای لیست پیوندی گرفته مطابق زیر است:

$1 \rightarrow 9 \rightarrow 3 \rightarrow 8 \rightarrow 4 \rightarrow 5 \rightarrow 3 \rightarrow 0 \rightarrow 3 \rightarrow 9$

همان طور که از خروجی پیداست، $node$ شامل عدد 2 از لیست پیوندی خارج شده و دیگر به آن دسترسی نداریم. پس می توان گفت که "memory leak" رخ داده است.

سؤال 3:



سؤال 4 :

A	B	C	output
0	0	0	0
1	0	0	0
0	1	0	0
0	0	1	0
1	1	0	1
1	0	1	1
0	1	1	1
1	1	1	1

[مدار شماره 1]

A	B	C	output
0	0	0	0
1	0	0	1
0	1	0	1
0	0	1	1
1	1	0	0
1	0	1	0
0	1	1	0
1	1	1	1

[مدار شماره 2]

سؤال 5 :

