

(1) با توجه به ثابت θ ، c_1 و c_2 و n_0 را به گونه ای پیدا کنیم که برای $n \geq n_0$

$$c_1 n^3 \leq 2n^3 + 2\omega n^2 \leq c_2 n^3$$

باید واضح است که برای $n \geq 1$

$$2n^3 \leq 2n^3 + 2\omega n^2 \leq 2\sqrt{n} n^3$$

(ب) اگر $f(n)$ را از $f(n)$ که از $f(n)$ و $g(n)$ است بدانیم که $n \geq n_0$ باشد یعنی c_1 و c_2 را به گونه ای

$$c_2 (f(n) + g(n)) \leq f(n) \leq c_1 (f(n) + g(n))$$

است با رعایت راست عبارت را بر می گیریم:

$$f(n) \leq 1 \times (f(n) + g(n))$$

$$c_1 = 1$$

$$g(n) \leq f(n)$$

حالا بیایم عبارت $g(n) \leq f(n)$ را فرض استوار داریم

$$g(n) + f(n) \leq f(n) + f(n)$$

$$\frac{1}{2} (g(n) + f(n)) \leq \frac{1}{2} (f(n) + f(n))$$

$$\frac{1}{2} (g(n) + f(n)) \leq f(n)$$

(1) آرایه: عناصر صف اولی آرایه در زیر ماتریس بصورت سطری و ستونی در یک طایفه قرار می گیرد.

void Transport (SparseMatrix A) (2)

 SparseMatrix AT;

 AT.row = A.col;

 AT.col = A.row;

 AT.Terms = A.Terms;

 int k=0;

 for (int i=0; i < AT.Terms; i++)

 { for (int j=0; j < AT.Terms; j++)

 if (A[j][i] == 0)

①

$$\left\{ \begin{aligned} AT [k][z] &= A[z][z]; // set value \\ AT [k][.] &= A[z][1]; // set row \\ AT [k][1] &= A[z][.]; // set col. \\ & k++; \end{aligned} \right\}$$

} }

(ع) اعتبار عبارت را بصورت کامل پراسترنواری و کسینر پس در عبارت سونی عملگر را بنویسید
 از پراسترنواری در مورد عبارت سونی عملگر را بنویسید قبل از پراسترنواری سینی

$$(((1 + ((2/4)/9)) - 5) + ((3 * 15)/3))$$

$$1 \ 2 \ 4 \ / \ 9 \ / \ + \ 5 \ - \ 3 \ 15 \ * \ 3 \ / \ +$$

$$+ \ - \ + \ 1 \ / \ * \ 2 \ 4 \ 9 \ 5 \ / \ * \ 3 \ 15 \ 3 \quad (ب)$$

$$((((A + B) * D) + (E / (F * (A * D)))) \quad (د)$$

$$AB + D * EFAD * + / + C +$$

$$+ + * + ABD / E + F * AD$$

(ع) = شروع

۱ تا ۴ تمام عبارت حاصل ۲ و ۳ را اوایل بنویسید
 ۲ کتب کار استر خوان در صورتی که کار استر عملیات آن را در دست درج کن
 ۳ در صورتی که عملیات در عملیات از دست POP که در آن قرار دارد عملیات
 عملیات پراسترنواری عبارت infid عملیات را جاد کن در دست درج کن
 ۴ عبارت سونی را POP که در دست کن

void myProgram (queue q)

{ int min = MAX_INT;

queue q2;

while (!q.isEmpty())

{ int a;

q.delete(a);

if (a < min)

min = a;

q2.insert(a);

}

while (!q2.isEmpty())

{ int a;

q2.delete(a);

q.insert(a);

}

cout << "min is" << min;

}

int GenList::depth (GenList * node, int s)

{ if (!s) return 0;

GenList * p = s;

int m = 0;

while (p)

{ if (p->tag)

{ int n = depth (p->dlink)

if (m < n) m = n;

}

p = p->link;

}

```
return m+1;
}
```

```
void sum_function (List *a, List *b, cv)
{
    int sum, carry = 0;
    List *c = new List();
    Node Node *last = 0;
```

```
ListNode *ptr = a->first;
```

```
ListNode *ptr2 = b->first;
```

```
while (ptr && ptr2)
```

```
{ Node * newnode = new Node;
```

```
sum = carry + (ptr->data-'0') +
            (ptr2->data-'0');
```

```
carry = sum / 10;
```

```
sum = sum % 10;
```

```
newnode->data = sum + '0';
```

```
newnode->link = last;
```

```
last = newnode;
```

```
}
```

```
while (ptr)
```

```
{ Node * newnode = new Node;
```

```
sum = carry + (ptr->data-'0');
```

```
carry = sum / 10;
```

```
sum = sum % 10;
```

```
newnode->data = sum + '0';
```

```
newnode->link = last;
```

```
last = newnode;
```

```
}
```

```
while (ptr2)
```

```
{ node * newnode = new node;
```

```
sum = carry + (ptr2->data - '0');
```

```
carry = sum / 10;
```

```
sum = sum % 10;
```

```
newnode->data = sum + '0';
```

```
newnode->link = last;
```

```
last = newnode;
```

```
}
```

```
};
```