Line	Code	Cost
1	**void** *Radix*::*SortMgr*(*istream*& ***i***,*ostream*& ***o***)	**0**
2	{ *List L*;	**0**
3	*L*.*Scan*(*i*);	**0**
4	*L*.*Print*(*o*,"**Unsorted List**");	**0**
5	**int** *p*=*L*.*Longest*()-1;	**2**
6	**while** (*p*>=0;){	p
7	*BucketSort*(*L*,*p*);	$p(13n+14b+2)$
8	*p*--;	p
9	}	**1**
10	*L*.*Print*(*o*,"**Sorted List**");	**0**
11	}	**0**
		$T(n) = p(13n+14b+2)+2p+3$

Line	Code	Cost
1	**void** *Radix*::*BucketSort*(***List***& *L*,**int** *p*)	**0**
2	{ *List B*[256];	**0**
3	**while** (!*L*.*IsEmpty*()) {	n
4	*Element** *e*=*L*.*HeadRemove*();	$4n$
5	**if** (*e*->*Length*()<*p*+1) {	$2n$
6	*B*[0].*TailInsert*(e);	$4n$
7	} **else** {	**0**
8	**char** *b*=*e*->*id*[*p*];	$2n$
9	*B*[*b*].*TailInsert*(*e*);	$4n$
10	}	**0**
11	}	**0**
12	**int** *a*=0;	**1**
13	**while** (*a*<256) {	b
14	*L*.*Join*(*B*[*a*]);	$12b$
14	*a*++;	b
16	}	**1**
17	}	**0**
		$T(n) = 13n+14b+2$

Line	Code	Cost
1	*Element* List*::*HeadRemove*(void)	0
2	{ ***Element**** *e=head*;	1
3	if (*head==tail*) {	1
4	*head=tail*=0;	2
5	} else {	0
6	*head=head->next*;	2
7	}	0
8	return *e*;	0
9	}	0
		$T(n) = 4$

Line	Code	Cost
1	void *List*::*TailInsert*(*Element* e*)	0
2	{ if (*head*) {	0
3	*tail->next=e*;	2
4	} else {	0
5	*head=e*;	1
6	}	0
7	*tail=e*;	1
8	*e->next*=0;	2
9	}	0
		$T(n) = 4$

Line	Code	Cost
1	void *List*::*Join*(*List*& *L*)	
2	{ if (*L.IsEmpty*() && *IsEmpty*()) return;	2
3	if (*L.IsEmpty*() && !*IsEmpty*()) return;	3
4	if (!*L.IsEmpty*() && *IsEmpty*()) {	3
5	*head=L.head*; *tail=L.tail*;	2
6	*cursor=L.cursor*; *longest=L.longest*;	2
7	*L.head=L.tail=L.cursor*=0; *L.longest*=0;	3 1
8	}	
9	if (!*L.IsEmpty*() && !*IsEmpty*()) {	4
10	*tail->next=L.head*;	3
11	*tail=L.tail*;	1
12	*L.head=L.tail=L.cursor*=0; *L.longest*=0;	3 1
13	}	
14	}	
		$T(n) = 12$