

上海交通大学试卷 (A 卷)

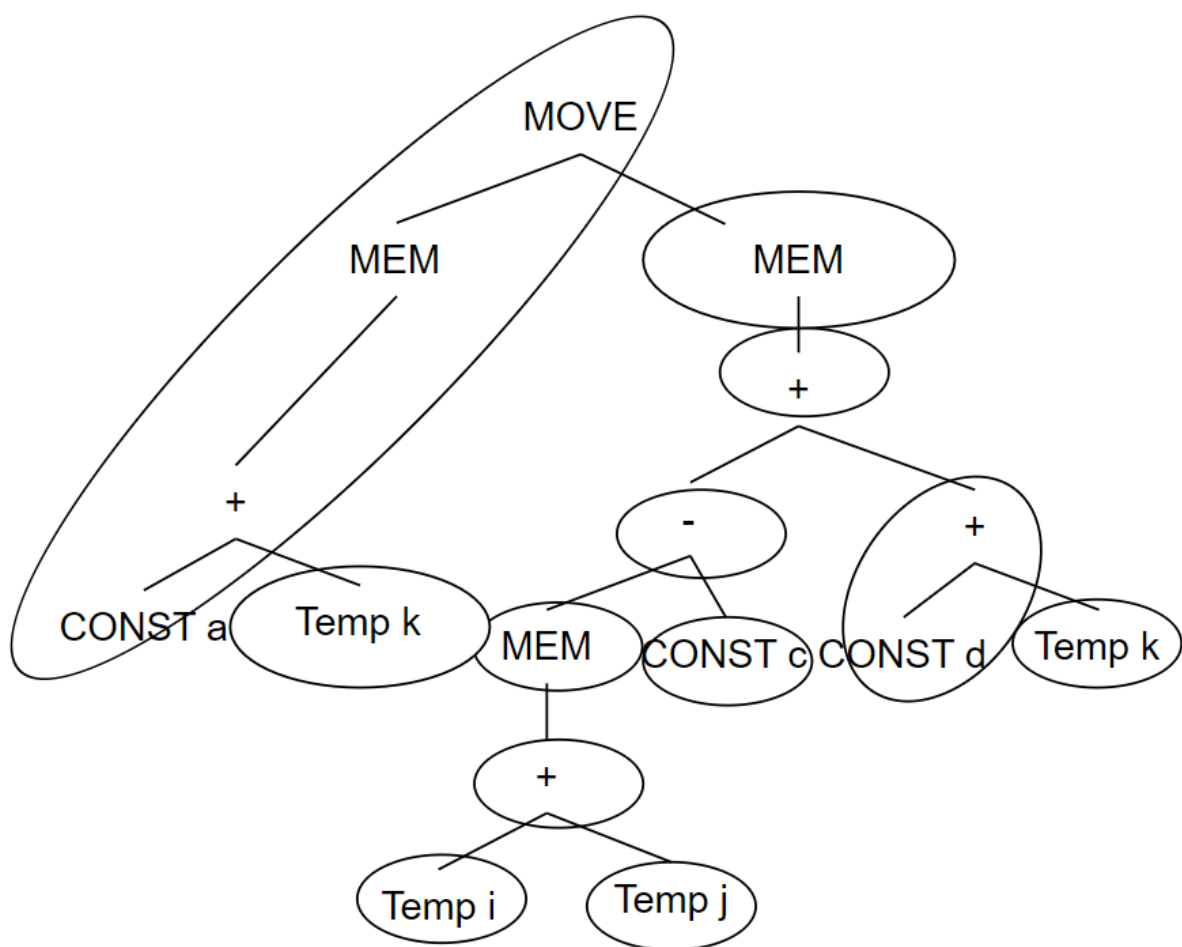
(2023 至 2024 学年 第 1 学期)

班级号_____ 学号_____ 姓名 _____

课程名称_____ 计算机系统基础 (汇编) _____ 成绩 _____

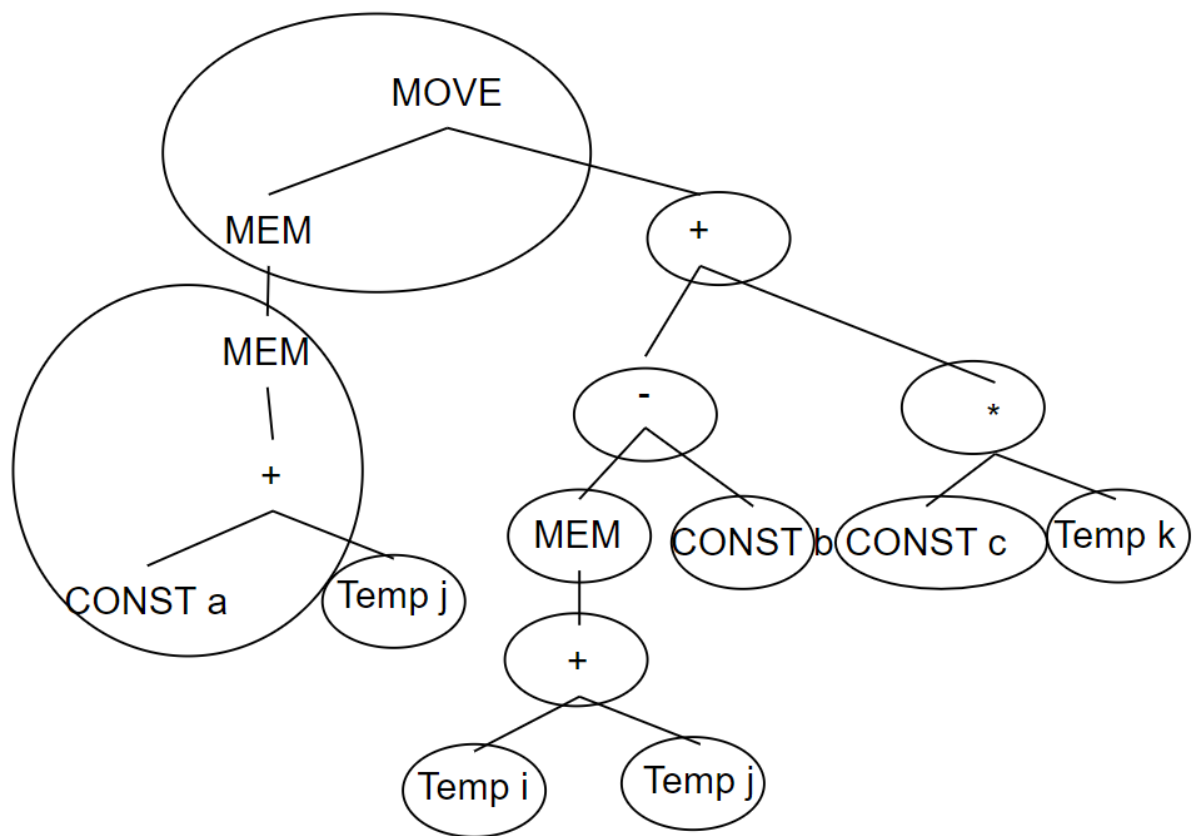
Problem 1: Instruction Selection

1.(12)



Cost: 28

2. (12)



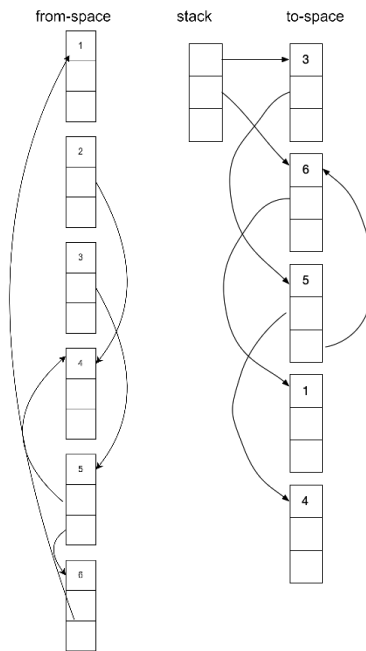
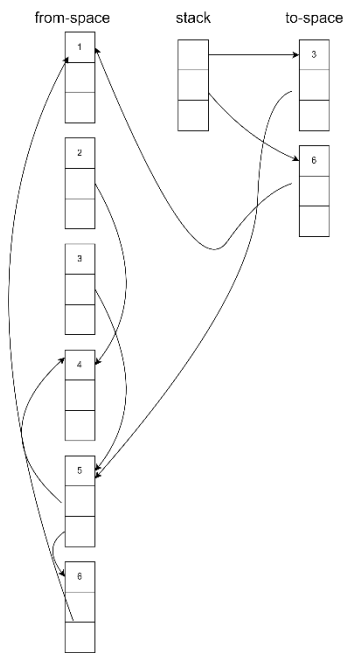
Cost: 30

Additional explanation:

1. Instruction must follow the format in effect, like $r1 \leftarrow r1 + c$ is not allowed.
2. Directly use $i/j/k$ to represent temp $i/j/k$ is not allowed.

Problem 2: Garbage Collection

1.



2.

- (1) Scanned: B C D
Survive: B C D E F G H
- (2) Scanned: E F G C D
Survive: E F G C D

3.

言之有理即可

Problem 3: Functional Programming

1.

不是。言之有理即可。

2.

`arr[0] = 9`

`arr[1] = 0`

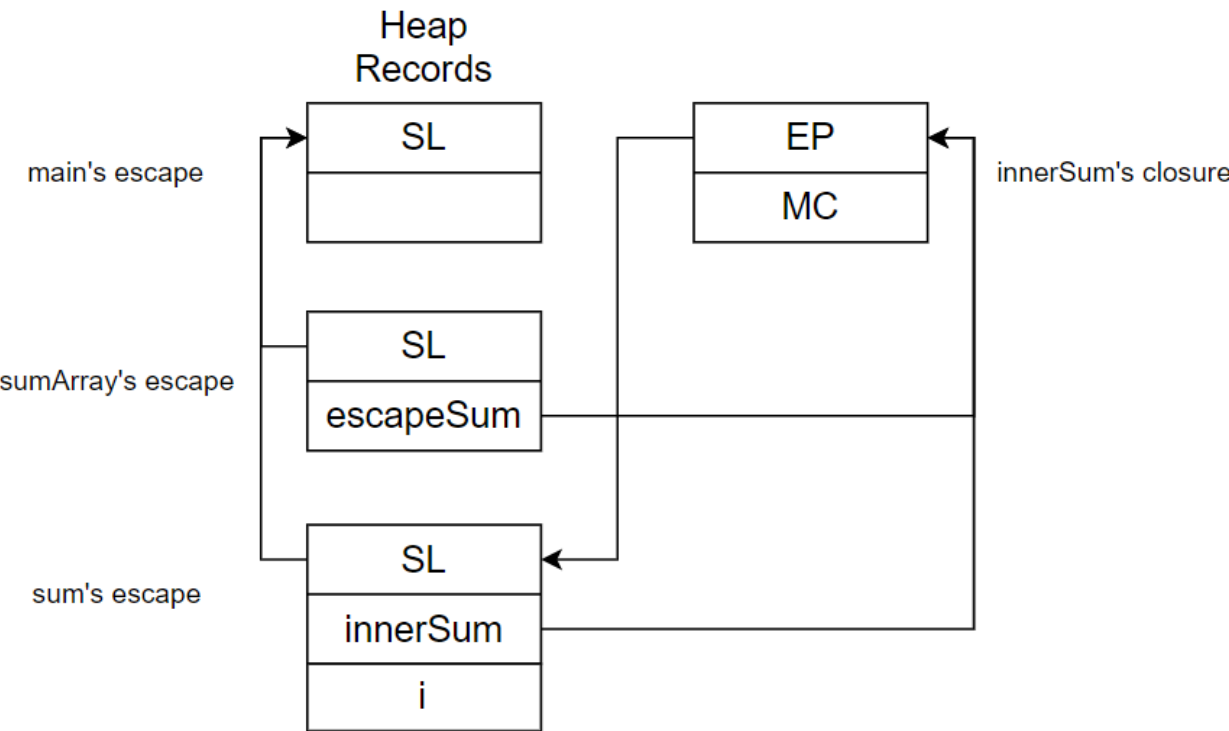
`arr[2] = 4`

`arr[3] = 4`

```
arr[4] = 9
```

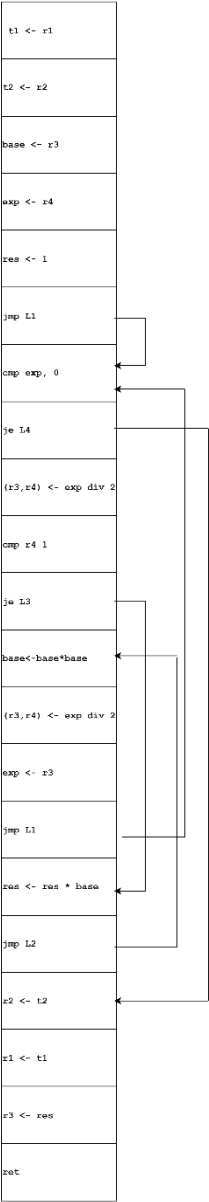
```
total = 28
```

3.



Problem 4: Register Allocation

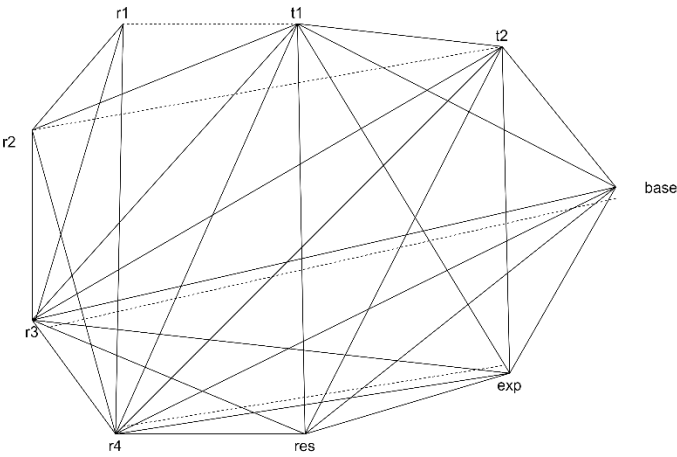
1.



2.

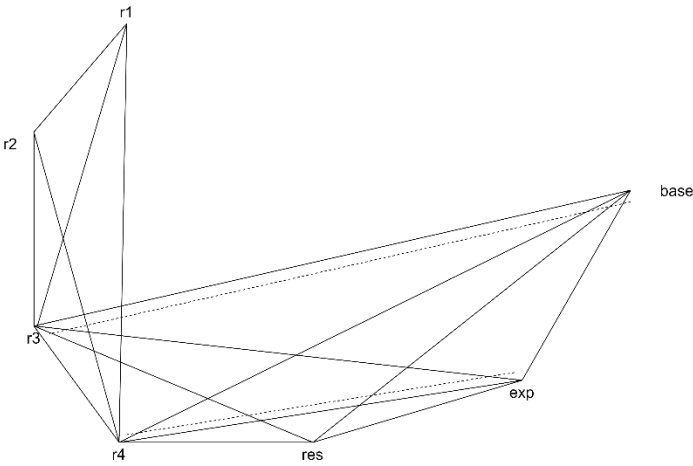
instr	def	use	in	out
t1 <- r1	t1	r1	r1 r2 r3 r4	t1 r2 r3 r4
t2 <- r2	t2	r2	t1 r2 r3 r4	t1 t2 r3 r4
base <- r3	base	r3	t1 t2 r3 r4	t1 t2 base r4
exp <- r4	exp	r4	t1 t2 base r4	t1 t2 base exp
res <- 1	res		t1 t2 base exp	t1 t2 base exp res
jmp L1			t1 t2 base exp res	t1 t2 base exp res
cmp exp, 0		exp	t1 t2 base exp res	t1 t2 base exp res
je L4			t1 t2 base exp res	t1 t2 base exp res
(r3,r4) <- exp div 2	r3 r4	exp	t1 t2 base exp res	t1 t2 base exp res r4
cmp r4 1		r4	t1 t2 base exp res r4	t1 t2 base exp res
je L3			t1 t2 base exp res	t1 t2 base exp res
base<-base*base	base	base	t1 t2 base exp res	t1 t2 base exp res
(r3,r4) <- exp div 2	r3 r4	exp	t1 t2 base exp res	t1 t2 base res r3
exp <- r3	exp	r3	t1 t2 base res r3	t1 t2 base exp res
jmp L1			t1 t2 base exp res	t1 t2 base exp res
res <- res * base	res	res base	t1 t2 base exp res	t1 t2 base exp res
jmp L2			t1 t2 base exp res	t1 t2 base exp res
r2 <- t2	r2	t2	res t1 t2	res t1 r2
r1 <- t1	r1	t1	res t1 r2	res r1 r2
r3 <- res	r3	res	res r1 r2	r1 r2 r3
ret			r1 r2 r3	\

3.

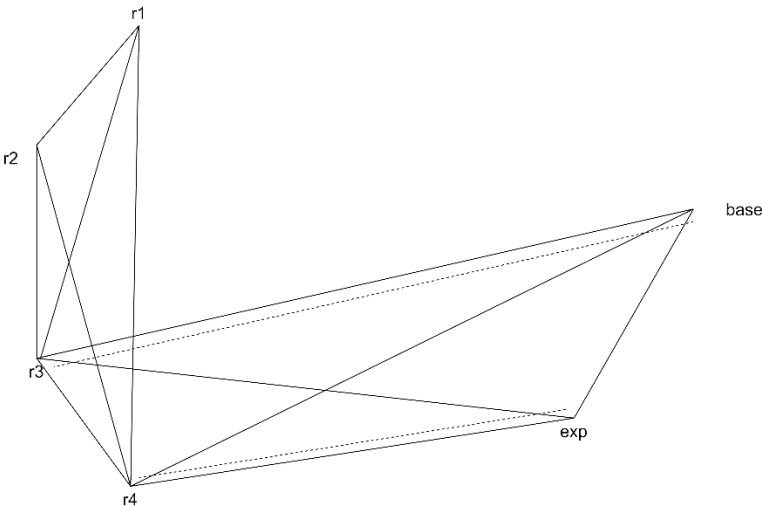


1. Spil:t1

2. Spil:t2



3. Spill res



4. Simplify exp(r2), base(r1)

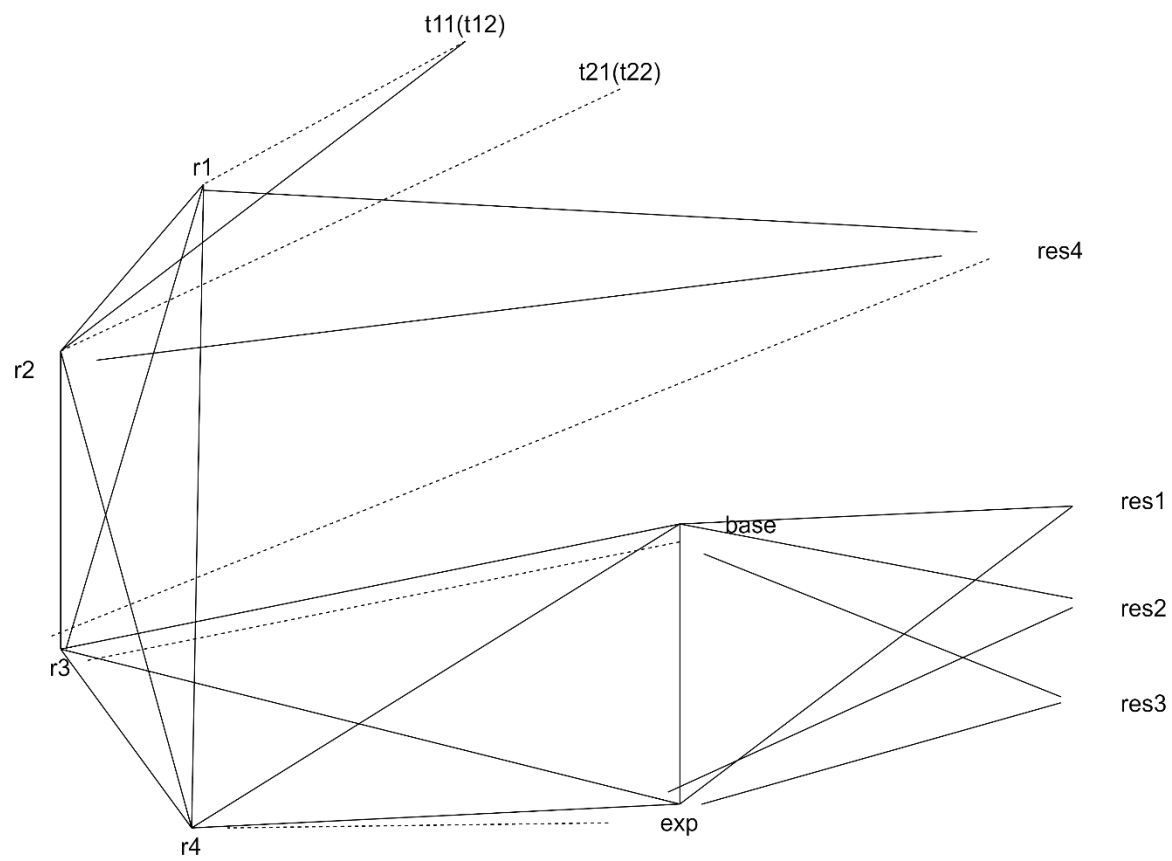
5. Select gap, res

6. Actually spill res, t2, t1

7. Def-use

instr	def	use	in	out
t11 <- r1	t11	r1	r1 r2 r3 r4	t11 r2 r3 r4
M[t1_loc] <- t11		t11	t11 r2 r3 r4	r2 r3 r4
t21 <- r2	t21	r2	r2 r3 r4	t21 r3 r4
M[t2_loc] <- t21		t21	t21 r3 r4	r3 r4
base <- r3	base	r3	r3 r4	base r4
exp <- r4	exp	r4	base r4	exp base
res1 <- 1	Res1		exp base	exp base res1
M[res_loc] <- res1			exp base res1	exp base
jmp L1			exp base	exp base
cmp exp, 0		exp	exp base	exp base
je L4			exp base	exp base
(r3,r4) <- exp div 2	r3 r4	exp	exp base	exp base r4
cmp r4 1		r4	exp base r4	exp base
je L3			exp base	exp base
base<-base*base	base	base	exp base	exp base
(r3,r4) <- exp div 2	r3 r4	exp	exp base	exp base r3
exp <- r3	exp	r3	exp base r3	exp base
jmp L1			exp base	exp base
Res2 <- M[res_loc]	Res2		exp base	exp base res2
Res3 <- res2 * base	Res3	Res2 base	exp base res2	exp base res3
M[res_loc]<-res3		res2	exp base res3	exp base
jmp L2			exp base	exp base
t22 <- M[t2_loc]	r2			t22
r2 <- t22			t22	r2

<code>t12 <- M[t1_loc]</code>	r1		r2	r2 t12
<code>r1 <- t12</code>			r2 t12	r1 r2
<code>res4 <- M[res_loc]</code>	Res4		r1 r2	r1 r2 res4
<code>r3 <- res4</code>	R3		r1 r2 res4	r1 r2 r3
<code>ret</code>			r1 r2 r3	\



- 8. Coalesce t11 -> r1
- 9. Coalesce t12 -> r1
- 10. Coalesce t21-> r2
- 11. Coalesce t22 -> r2
- 12. Coalesce res4 -> r3
- 13. Select res1 -> r2
- 14. Select res2 -> r2
- 15. Select res3 -> r2

```

quick_pow:
M[t1_loc] <- r1
M[t2_loc] <- r2
base <- r3
exp <- r4
r2 <- 1
M[res_loc] <- r2
jmp L1
L1:
cmp exp, 0
je L4
(r3,r4) <- exp div 2
cmp r4 1
je L3
L2:
base<-base*base
(r3,r4) <- exp div 2
exp <- r3
jmp L1
L3:
r2 <- M[res_loc]
r2 <- r2 * base
M[res_loc]<-r2
jmp L2
L4:
t22 <- M[t2_loc]
r2 <- t22
t12 <- M[t1_loc]
r1 <- t12
r3 <- M[res_loc]
ret

```