

Sistemi di Calcolo (A.A. 2014-2015)

Corso di Laurea in Ingegneria Informatica e Automatica
Sapienza Università di Roma

Esercizi riepilogativi sulla seconda parte del Modulo I – Tecniche di ottimizzazione

Domanda 1

Riscrivere il seguente frammento di programma C applicando manualmente constant folding:

```
double circonferenza(double raggio) {
    double PI = 3.14;
    return 2*PI*raggio;
}
```

Domanda 2

Riscrivere il seguente frammento di programma applicando manualmente loop-invariant code motion:

```
void tolower(char* s) {
    int i;
    for (i=0; ; i++) {
        int c = strlen(s);
        if (i >= c) break;
        s[i] = tolower(s[i]);
    }
}
```

Domanda 3

Riscrivere il seguente frammento di programma C applicando manualmente function inlining:

```
int abs(int x, int y) { return x > y ? x-y : y-x; }

int dist(int* v, int n) {
    int i, max=0;
    for (i=1; i<n; i++)
        if (abs(v[i-1], v[i]) > max) max = abs(v[i-1], v[i]);
    return max;
}
```

Domanda 4

Riscrivere il seguente frammento di programma C applicando manualmente common subexpression elimination:

```
double dist(double x1, double y1, double x2, double y2) {
    return sqrt((x1-x2)*(x1-x2)+(y1-y2)*(y1-y2));
}
```

Domanda 5

Riscrivere il seguente frammento di programma C applicando manualmente strength reduction:

```
int f(x) {
    return x*5;
}
```

Domanda 6

Riscrivere il seguente frammento di programma C ottimizzando lo spazio richiesto dalla struttura:

```
struct S {
    unsigned char eta;
    int matricola;
    unsigned short data_nascita;
    char* nome;
};
```

Quanti byte occupa la struttura prima e dopo l'ottimizzazione? Assumere un sistema a 32 bit conforme con le convenzioni della System V ABI.

Domanda 7

Si consideri il seguente frammento di programma C e la sua traduzione in IA32:

<pre>void f() { int a, b, c; g(&a, &b, &c); }</pre>	<pre>f: subl \$44, %esp leal 28(%esp), %eax movl %eax, 8(%esp) leal 24(%esp), %eax movl %eax, 4(%esp) leal 20(%esp), %eax movl %eax, (%esp) call g addl \$44, %esp ret</pre>
---	---

Perché il compilatore alloca 44 byte nello stack frame di f? I 44 byte allocati sono tutti utilizzati?

Domanda 8

Si consideri il seguente frammento di programma C e la sua traduzione in IA32:

<pre>int f(int x) { if (0) return 2*x; return 3*x; }</pre>	<pre>f: movl 4(%esp), %eax leal (%eax,%eax,2), %eax ret</pre>
--	---

Quali delle seguenti ottimizzazioni sono state applicate dal compilatore?

A	Dead code elimination	B	Function inlining
C	Constant folding	D	Strength reduction

Domanda 9

Si consideri il seguente frammento di programma C e la sua traduzione in IA32:

<pre>void f(int* v, int n, int x) { while (--n >= 0) v[n] = x*x; }</pre>	<pre>f: movl 4(%esp), %edx movl 12(%esp), %ecx movl 8(%esp), %eax subl \$1, %eax js L1 imull %ecx, %ecx L3: movl %ecx, (%edx,%eax,4) subl \$1, %eax cmpl \$-1, %eax jne L3 L1: ret</pre>
---	---

Quali delle seguenti ottimizzazioni sono state applicate dal compilatore?

A	Dead code elimination	B	Hoisting
C	Constant folding	D	Strength reduction

Domanda 10

Si consideri il seguente frammento di programma C e la sua traduzione in IA32:

<pre>int f(int x, int y) { return (x-y)*(x-y); }</pre>	<pre>f: movl 4(%esp), %eax subl 8(%esp), %eax imull %eax, %eax ret</pre>
--	--

Quali delle seguenti ottimizzazioni sono state applicate dal compilatore?

A	Dead code elimination	B	Function inlining
C	Common subexpression elimination	D	Strength reduction

Domanda 11

Si consideri il seguente frammento di programma C e la sua traduzione in IA32:

<pre>char sgn(int x) { return x > 0; } char f(int x, int y) { return sgn(x-y); }</pre>	<pre>sgn: cmpl \$0, 4(%esp) setg %al ret f: movl 4(%esp), %eax subl 8(%esp), %eax testl %eax, %eax setg %al ret</pre>
--	--

Quali delle seguenti ottimizzazioni sono state applicate dal compilatore?

A	Dead code elimination	B	Function inlining
C	Common subexpression elimination	D	Constant folding

Domanda 12

Quale speedup ci aspettiamo per un programma se ottimizziamo di un fattore 1.5x una sua porzione che richiede il 60% del tempo di esecuzione?

Domanda 13

Qual è lo speedup massimo ottenibile per un programma se ottimizziamo una sua porzione che richiede il 30% del tempo di esecuzione?

Domanda 14

Conviene rendere 2 volte più veloce una funzione che richiede il 10% del tempo di esecuzione, oppure velocizzare del 10%¹ una funzione che richiede il 90% del tempo di esecuzione?

¹ Velocizzare una funzione A del 10% significa che $T'_A = T_A \cdot 0.9$, quindi uno speedup per A pari a $\frac{T_A}{T'_A} = \frac{1}{0.9} = 1.11x$.