

- 【解答】
- 〔設問1〕 aーオ
- 〔設問2〕 bーア, cーウ, dーイ
- 〔設問3〕 イ

【解説】

プログラムの並列実行をテーマとした問題は、基本情報処理技術者試験では初めての出題である。一見、難しく感じられるかもしれないが、問題文の説明に沿って考えれば、解答できる問題である。

〔設問1〕

設問1は高速化率の式に値を代入して計算すれば、容易に解答できる問題である。図1のプログラムAにおいて、データ作成、計算処理、結果出力の処理時間の割合が7:90:3の場合、単一のCPUで実行したときと比べた高速化率を5以上にするには、CPUが最低何個必要であるかが問われている。

問題文の冒頭部分と図1から、データ作成は並列実行ができず、計算処理は並列実行が可能、そして結果出力も並列実行ができないことが分かる。したがって、計算処理の部分だけがスレッド化によるマルチプロセッサにおけるスレッド実行が可能で、その部分は全体の $\frac{90}{100}$ であるため $r=0.9$ と導くことができる。

高速化率の式は、問題文にあるとおり、式1になる。

$$E = \frac{1}{1-r + \left(\frac{r}{n}\right)} \quad \cdots \text{式1}$$

ただし

n: CPUの数 ($n \geq 1$)

r: 対象とする処理のうち、並列実行が可能な部分の処理時間の割合 ($0 \leq r \leq 1$)

ここで、目標とする高速化率Eは、5以上であるから、式1に代入すると

$$5 \leq \frac{1}{1-0.9 + \frac{0.9}{n}} \quad \cdots \text{式2}$$

$$5 \times (1-0.9 + \frac{0.9}{n}) \leq 1$$

$$5 - 4.5 + (5 \times \frac{0.9}{n}) \leq 1$$

$$0.5 + \frac{4.5}{n} \leq 1$$

$$\frac{4.5}{n} \leq 0.5$$

$$\frac{4.5}{n} \leq 0.5n$$

$$n \geq 9$$

となる。したがって、(オ)が正解である。

ちなみに式1はアムダールの法則と呼ばれる有名な式であり、応用情報処理技術者試験などではよく出題される問題である。この問題では、 $r=0.9$ で計算しているが、実際にはもっと低いことが多い。 r の値が低いと、CPUの数を増やしても効率的に処理を向上させることは難しくなる。これは、マルチプロセッサシステム以外にも、現在のパソコンのCPUでよく使われるマルチコアプロセッサでも同様であることに注意したい。

〔設問2〕

設問2は、プログラムを分割してスレッド化した場合において、並列実行が可能なループの問題である。図3の並列実行が可能なループの例を見ると、元のプログラムはループ変数*i*が1から100まで繰り返すループであり、**a[i]**を更新している。それを4個のスレッド(スレッド1〜スレッド4)に分割し、ループ変数*i*の範囲は以下の様に分けられている。

- ・スレッド1は、ループ変数*i*が1から25まで繰り返す
- ・スレッド2は、ループ変数*i*が26から50まで繰り返す
- ・スレッド3は、ループ変数*i*が51から75まで繰り返す
- ・スレッド4は、ループ変数*i*が76から100まで繰り返す

各スレッドの*i*の範囲は別であるため、**a[i]**を更新する処理も独立している。そのため、並列実行が可能である。

- ・空欄b: プログラム1を見ると、ループ変数*i*が2から*n*まで繰り返すループであり、**a[i]**の更新処理である**a[i]←a[i-1]**を含んでいる。具体例を挙げると*i*=24とその次の*i*=25のときは、**a[24]←a[23]+b[24]**、**a[25]←a[24]+b[25]**となり、これは、解答群(ア)の「更新した値が次の繰返しで参照されるので」に当たる。もし、先ほど示したスレッドの分割のように、スレッド1は、ループ変数*i*が1から25まで繰り返し、スレッド2は、ループ変数*i*が26から50まで繰り返すように分割した場合、**a[24]←a[23]+b[24]**の処理はスレッド1、**a[25]←a[24]+b[25]**の処理はスレッド2と別スレッドになる。その場合、スレッド1で**a[24]←a[23]+b[24]**の処理が行われる前に、スレッド2で**a[25]←a[24]+b[25]**の処理が行われる可能性が高く、データの更新と参照の順番が変わる可能性が高い。そのため、このプログラム1は、並列実行できない。したがって、(ア)が正解である。
- ・空欄c: プログラム2を見ると、ループ変数*i*が1から*n*まで繰り返すループであり、**a[i]**の更新処理である**a[i]←a[i+1]**を含んでいる。具体例を挙げると*i*=24とその次の*i*=25のときは、**a[24]←a[25]+b[24]**、**a[25]←a[26]+b[25]**となり、これは、解答群(ウ)の「参照した値が次の繰返しで更新されるので」に当たる。空欄bで考察したように、スレッド1は、ループ変数*i*が1から25まで繰り返し、スレッド2は、26から50まで繰り返すように分割した場合、**a[24]←a[25]+b[24]**の処理はスレッド1、**a[25]←a[26]+b[25]**の処理は、スレッド2と別のスレッドになる。その場合、スレッド1で**a[24]←a[25]+b[24]**の処理が行われる前に、スレッド2で**a[25]←a[26]+b[25]**の処理が実行される可能性が高く、データの更新と参照の順番が変わる可能性が高い。そのため、このプログラム2も、並列実行できない。したがって、(ウ)が正解である。

- ・空欄d: プログラム3を見ると、ループ変数*i*が1から*n*まで繰り返すループであり、**a[i]**の更新処理である**a[i]←a[i+m]**を含んでいる。この場合は、*m*の値によって、条件が変わってくる。
まずは、dに関する解答群(ア) $m \geq 0$ の場合を当てはめて考えてみる。例えば*m*=1の場合は、**a[i]←a[i+1]+b[i]**となり、プログラム2と同じになり、並列実行できない。
次に(イ) $m \geq n$ の場合を当てはめて考えてみる。例えば*n*=100であり、*m*=100の場合は、**a[i]←a[i+100]+b[i]**となり、最初の*i*が1のときは**a[1]←a[101]+b[1]**となり、最後の*i*が100のときは、**a[100]←a[201]+b[100]**となる。この場合、各処理は独立しておりデータの更新と参照の順番が変わる可能性はない。よって、並列実行が可能である。
最後の(エ) $m \leq n$ の場合を当てはめて考えてみると、この場合は、*m*=1の場合もあり得るのでプログラム2と同じになり、並列実行できない。
よって、(イ)の $m \geq n$ であることが保障されていれば、並列実行が可能となる。したがって、(イ)が正解である。

〔設問3〕

プログラム4では、**a[ip[i]]**の式となり、配列**a**で更新する要素を示すインデックスの値が配列**ip**で間接的に決定される。そのため、**ip[i]**の値によっては、並列実行される複数のスレッドから値が格納される配列**a**の要素番号が重複するため、スレッド間の処理順序の違いによって、最終的な配列**a**の内容が変わってしまうため、並列実行できない場合もある。

例えば、解答群の(ア)の場合、**ip[i]**の値が1,2,3,4,5,1,2,3,4,5,1…と同じ値が繰り返して出現している。そのため、**a[ip[i]]**も、**a[1]**、**a[2]**、**a[3]**、**a[4]**、**a[5]**、**a[1]**、**a[2]**、**a[3]**、**a[4]**、**a[5]**…と重なることになる。具体的に言うと、**a[1]**の場合は、各スレッドによって**a[1]←b[1]**、**a[1]←b[6]**、**a[1]←b[11]**、**a[1]←b[16]**の処理が行われることになり、データの更新と参照の順番が変わると最終的な配列**a**の内容が変わってしまうので、並列実行できない。

解答群の(イ)の場合、**ip[i]**の値がランダムであり、重複がないため、複数のスレッドが同じ**a[i]**に代入することはなく、処理順序の違いがあっても、最終的な配列**a**の内容に違いはないと考えられる。したがって、(イ)が正解である。

ちなみに(ウ)、(エ)も重複する**ip[i]**の値が出現するため、データの更新と参照の順番が変わると最終的な配列**a**の内容が変わるため、不適切である。