

MA'LUMOTLAR TUZILISHI VA PYTHON DASTURLASH TILIDA ALGORITMIK FIKRLASH JARAYONLARI.

S.D. Raximov

*“Informatika va kompyuter grafikasi” kafedrası assistenti
Toshkent davlat transport universiteti, O‘zbekiston*

Anatatsiya: *Ma'lumotlar tuzilishida algoritmlar, ularning belgilarini, munosabatlarini tahlil qilish va iloji boricha ko'proq muammolarni hal qilishning ahamiyatini tushuntirishdir. Algoritmning asosiy elementlarini, algoritm tahlilining ahamiyatini tushunishga e'tibor qarataylik va keyin yuqorida aytib o'tilganidek, asta-sekin boshqa mavzularga o'tamiz. Uning bo'limini tugatgandan so'ng, siz algoritm (rekursiv funksiyalar) berilgan birlikning murakkabligini topa olamiz.*

Kalit so'zlar: *data, data type, integer, float, Primitiv, primitive data types, lions, protsedura, eksponensial.*

Ma'lumotlar turlari O'zgaruvchilar ta'rifiga o'tishdan oldin ularni eski matematik tenglamalar bilan bog'laymiz. Har birimiz bolaligimizdan beri ko'plab matematik tenglamalarni yechdik. Misol tariqasida quyidagi tenglamani ko'rib chiqamiz:

$$x^2 + 2y - 2 = 1$$

Biz bu tenglamadan foydalanish haqida tashvishlanmaymiz. Biz tushunishimiz kerak bo'lgan muhim narsa shundaki, tenglamaning nomlari (x va y) mavjud bo'lib, ular qiymatlarni (data) saqlaydi. O'rtacha ma'lumotlarni ifodalovchi nomi (x va y) va to'ldiruvchisi. Shunga o'xshab, kompyuter orqali dasturlashda bizga ma'lumotlarni saqlash uchun biror o'zgaruvchi kerak va o'zgaruvchilar orqali ifodalash yo'lidir yo'lidir.

Yuqorida aytib o'tilgan tenglamada x va y o'zgaruvchilari integral sonlar (10, 20), haqiqiy sonlar (0.23, 5.5) yoki faqat 0 va 1 kabi har qanday qiymatlarni qabul qilishi mumkin. Tenglamani yechish uchun biz ularni bog'lashimiz kerak. Ular qanday qiymatlarni qabul qilishlari mumkinligini ko'ring va Data type bu maqsad uchun dasturlashda kompyuter orqali qo'llaniladigan nomdir.

Dasturlash tilidagi ma'lumotlar turi - bu oldindan belgilangan qiymatlarga ega bo'lgan ma'lumotlar to'plami. Dasturiy til tipi turlariga misollar: butun son, belgi nuqtasi, birlik n raqam, belgi, satr va boshqalar.

Kompyuter xotirasi nol va birlar bilan to'ldirilgan. Agar bizda muammo bo'lsa va biz uni kodlamoqchi bo'lsak, nol va birliklar nuqtai nazaridan yechimni ta'minlash juda qiyin. Foydalanuvchilarga yordam berish uchun dasturlash tillari va kompilyatorlar bizga ma'lumotlar turlarini taqdim etadi. Masalan, int (integer) 2 baytni oladi (haqiqiy qiymat kompilyatorga bog'liq), float 4 baytni oladi va hokazo. Bu shuni anglatadiki, biz xotirada 2 baytni (16 bit) birlashtiramiz va uni butun son deb ataymiz. Xuddi shunday,

4 baytni (32 bit) birlashtirib, uni float sifatida yig'ish. Ma'lumotlar turi kodlash harakatini kamaytiradi. Ma'lumotlar past darajasida tiplarning ikki turi mavjud:

- Tizim tomonidan aniqlangan ma'lumotlar turlari (uni Primitiv ma'lumotlar turlari deb ham ataladi)

- Foydalanuvchi tomonidan belgilangan ma'lumotlar turlari.

Tizim tomonidan belgilangan ma'lumotlar turlari (Primitive data types). Tizim yoyi bilan belgilanadigan ma'lumotlar turlari ibtidoiy ma'lumotlar turlari deb ataladi. Ko'pgina dasturlash tillari tomonidan taqdim etiladigan ibtidoiy ma'lumotlar turlari quyidagilardir: int, noat, char, double, boo!, va boshqalar. Har bir ibtidoiy ma'lumotlar turi uchun ajratilgan bitlar soni dasturlash moslamalariga, kompilyatorga va operatsion tizimga bog'liq. Demak bir xil ibtidoiy ma'lumotlar turi bo'lsa, agar tillar turli o'lchamlardan foydalanishi mumkin. Ma'lumot turlarining o'lchamiga qarab, mavjud qiymatlarda (domen) ham o'zgaradi. Misol uchun, "int" 2 bayt yoki 4 bayt bo'lishi mumkin. Agar u 2 baytni (16 bit) olsa, unda umumiy mumkin bo'lgan qiymatlar minus 32,768 dan plyus 32,767 gacha (-215 dan 215-1 gacha). Agar u 4 bayt (32 bit) olsa, mumkin bo'lgan qiymatlar - 2,147,483,648 va +2,147,483,647 (-231 dan 2:11 - 1) oralig'ida bo'ladi. Boshqa ma'lumotlar turlari bilan ham xuddi shunday t.hc holati.

Foydalanuvchi tomonidan belgilangan ma'lumotlar turlari Agar tizim tomonidan aniqlangan ma'lumotlar turlari etarli bo'lmasa, ko'pgina dasturlash tillari foydalanuvchilarga o'zlarining ma'lumotlar turlarini, foydalanuvchi tomonidan belgilangan ma'lumotlar turlarini aniqlashga imkon beradi. Foydalanuvchi ma'lumotlar turlarining yaxshi namunalari yoy: C/C ++ va Java-dagi yangi tuzilmalar. Misol uchun, quyida keltirilgan parchada biz tizim tomonidan aniqlangan ko'plab ma'lumotlar turlarini birlashtiramiz va foydalanuvchi tomonidan belgilangan ma'lumotlar turini "newType" nomi bilan chaqirish. Bu kompyuter xotirasi bilan ishlashda ko'proq moslashuvchanlik va qulaylik beradi.

Struct newType

```
{  
Int data 1;  
float data 2;  
.....  
char data;  
}
```

Yuqoridagi muhokamaga asoslanib, biz o'zgaruvchilar bo'yicha ma'lumotlarga ega bo'lganimizdan so'ng, biz buni aniqlash uchun qandaydir mexanizmga muhtojmiz. ma'lumotlar muammolarni hal qiladi. Ma'lumotlar tuzilishi - bu kompyuterda ma'lumotlarni yig'ish va tartibga solishning o'ziga xos usuli unumli foydalanish mumkin. Ma'lumotlar strukturasi ma'lumotlarni tashkil qilish va saqlash uchun maxsus formuladir. Umumiy struktura turlariga massivlar, fayllar, bog'langan ro'yxatlar, steklar, navbatlar, ln.:cs, gra phs va boshqalar kiradi.

➤ chiziqli ma'lumotlar tuzilmalari: elementlarga kirish ketma-ketlikda amalga oshiriladi, ammo barchasini saqlash shart emas, elementlar ketma-ket. Misollar: Bog'langan ro'yxatlar, barcha navbatlar to'plami.

➤ chiziqli bo'lmagan ma'lumotlar tuzilmalari: Ushbu ma'lumotlar strukturasi elementlari chiziqli bo'lmagan tartibda saqlangan. Misollar: daraxtlar va grafiklar.

Mavhum ma'lumotlar turlari (ADTs) ni aniqlashdan oldin, har xillarini ko'rib chiqaylik. tizim tomonidan belgilangan ma'lumotlarning ko'inishi. Biz bilamizki, sukut bo'yicha barcha ibtidoiy ma'lumotlar turlari (int, float va boshqalar) qo'llab-quvvatlanadi. qo'shish va ayirish kabi asosiy amallar. Tizim ibtidoiy ma'lumotlar turlari uchun amaliy dasturlarni taqdim etadi. Foydalanuvchi tomonidan belgilangan ma'lumotlar turlari uchun biz e operatsiyalarni ham aniqlashimiz kerak. Ushbu operatsiyalarni amalga oshirish biz ulardan haqiqatda foydalanmoqchi bo'lganimizda amalga oshirilishi mumkin. Bu degani, umuman olganda, foydalanuvchi tomonidan belgilangan ma'lumotlar turlari, ularning operatsiyalari bilan birga belgilangan.

Muammoni hal qilish jarayonini anglatish uchun biz ma'lumotlar sub'ektlarini ularning operatsiyalari bilan birlashtiramiz va biz chaqiramiz bu mavhum ma'lumotlar Ma'lumotlar turlari (ADTs). ADT ikkita pnrtdan iborat:

1. Ma'lumotlar deklaratsiyasi
2. "Lions" operasining deklaratsiyasi

Keng qo'llaniladigan ADT larga quyidagilar kiradi: Bog'langan ro'yxatlar, steklar, navbatlar, ustuvor navbatlar, ikkilik daraxtlar, dic lug'atlar, ajratilgan to'plamlar (Union va Find), xesh jadvallari, grafiklar va boshqa ko'plab rs. Masalan, stek dala tuzilmalarida ma'lumotlarni saqlashda LIFO (Last-InFirst-Out) mexanizmidan foydalanadi. Stakka kiritilgan oxirgi element bu o'chiriladigan birinchi element. Uning keng tarqalgan operatsiyalari quyidagilardir: stek yaratish, elementni ustiga surish stek, stekdan element chiqarish, stekning joriy yuqori qismini topish, stekdagi elementlar sonini topish stek va boshqalar. ADTlarni belgilashda bunday emas. amalga oshirish tafsilotlari haqida qayg'uring. Ular rasmga faqat biz ulardan foydalanmoqchi bo'lganimizda kiradi. Turli. ADT turlari har xil turdagi ilovalarga mos keladi va ba'zilar aniq vazifalarga juda mos keladi. Ushbu kitobning oxiriga kelib, biz ko'plab fanlarni ko'rib chiqamiz va siz ma'lumotlar tuzilmalarini ular hal qiladigan muammolar turi bilan bog'lash imkoniyatiga ega bo'lasiz.

Keling, pishiriq tayyorlash muammosini ko'rib chiqaylik. Tuxumdonni tayyorlash uchun biz quyidagi amallarni bajaramiz:

- 1) birir nimani qovurmoqchisiz.
- 2) yog 'ni oling.
 - a. bizda yoqilg'i bormi?
 - i. ha bo'lsa, skavarotkaga quying
 - ii. agar yo'q bo'lsa, biz yoqilg'i sotib olishni xohlaymizmi?

1. ha bo'lsa, tashqariga chiqing va sotib oling
2. agar yo'q bo'lsa, biz tugatishimiz mumkin.
- 3) Pechni yoqing va davom ettiring

Biz nima qilyapmiz, ma'lum bir muammo uchun biz bosqichma-bosqich protsedurani taqdim etamiz. Uni hal qilish. Algoritmning rasmiy ta'rifi quyidagicha bo'lishi mumkin:

Algoritm - bu berilgan muammoni hal qilish uchun bosqichma-bosqich ko'rsatmalar.

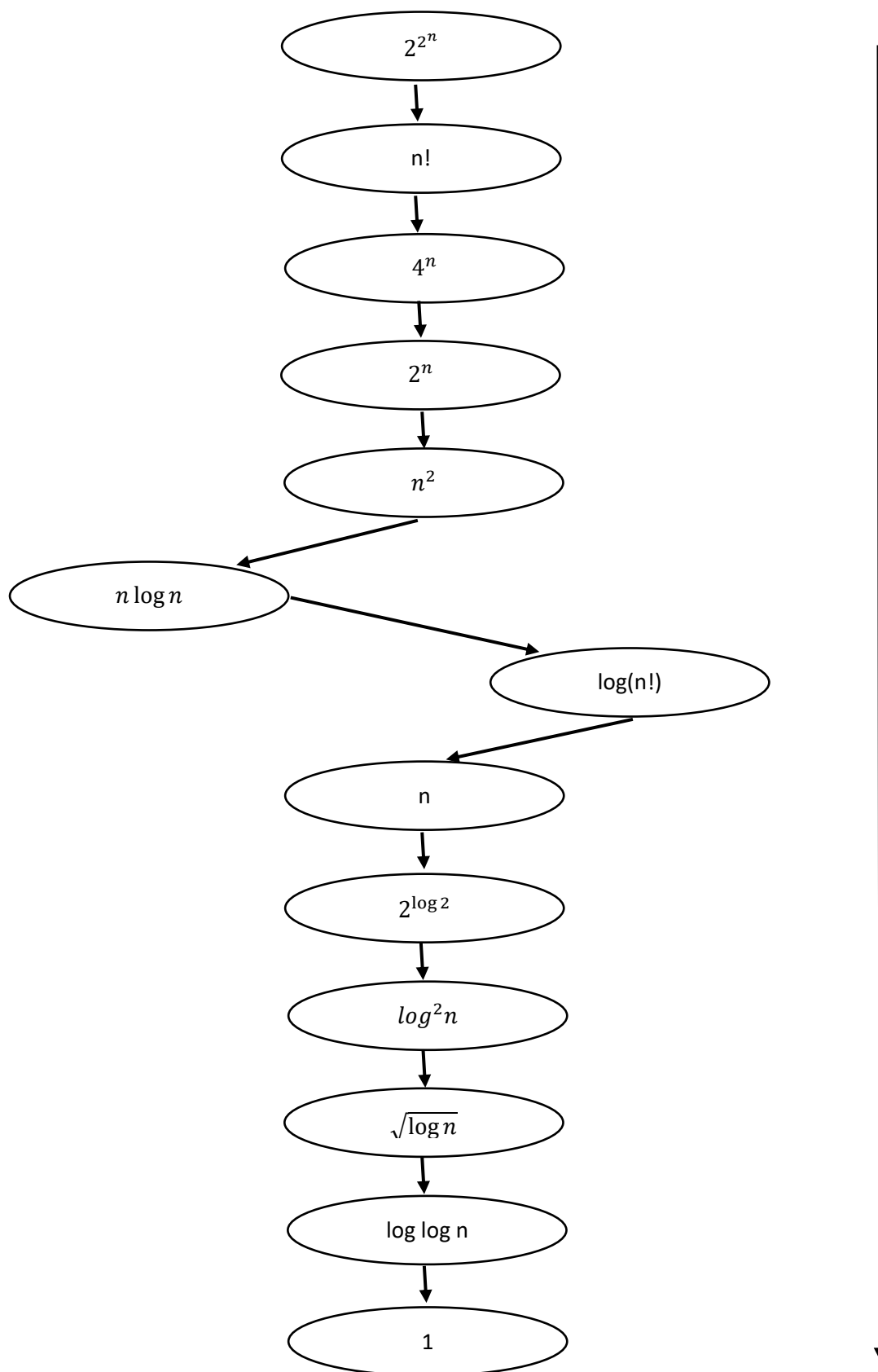
Lekin algoritmning har bir bosqichini isbotlay olmaymiz.

Algoritmni tahlil qilishda "A" dan "8" shaharga borish uchun buni amalga oshirishning ko'plab usullari bo'lishi mumkin: parvoz, avtobus, poezd va shuningdek velosipedda. Mavjudligi va qulayligiga qarab, biz o'zimizga mos keladiganini tanlaymiz. Xuddi shunday, bir xil muammoni hal qilish uchun bir nechta algoritmlari mavjud (masalan, saralash muammoning ko'plab algoritmlari mavjud, masalan, qo'shish tartibi, tanlash tartibi, tez tartiblash va boshqalar). Algoritm tahlili vaqt va makon bo'yicha qaysi algoritm eng samarali ekanligini aniqlashga yordam beradi.

Quyida keyingi boblarda duch keladigan o'sish sur'atlari ro'yxati keltirilgan.

Vaqtning murakkabligi	Nomi	Misol
1	Constanl	Bog'langan Iislning old qismiga klement qo'shish
$\log n$	Logarithmic	Tartiblangan massivdagi elementni topish
n	Chiziqli	Ovozsiz massivda elementni topish
$n \log n$	Chiziqli logarithmic	n ta element bo'yicha saralash
n^2	Kvadrat	Grafikdagi ikkita tugun orasidagi eng qisqa yo'l
n^3	Kub	Matritsalarini ko'paytirish
2^n	Eksponensial	Xanoy minoralari muammosi

Quyidagi diagrammada turli o'sish sur'atlari o'rtasidagi bog'liqlik ko'rsatilgan.
O'sish sur'atlarining pasayishi algoritmi.



Berilgan algoritmni o'zgartirish uchun biz algoritmni qaysi kirishlar bilan ko'proq vaqt sarflashini bilishimiz kerak (bajarish yaxshi) va qaysi kirishlar bilan algoritm uzoq vaqt talab etadi. Biz allaqachon algoritm bo'lishi mumkinligini ko'rdik ifoda shaklida ifodalanadi.

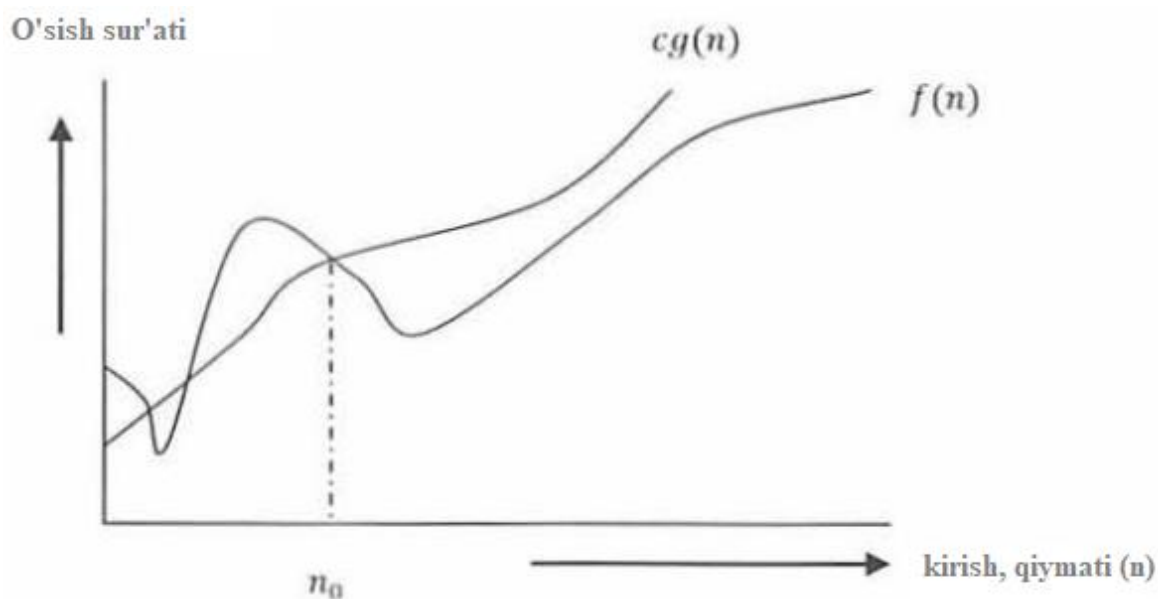
Bu shuni anglatadiki, biz algoritmni bir nechta ifodalar bilan ifodalaymiz. Bu kamroq vaqt talab qiladigan holatda va boshqasi uchun ko'proq vaqt talab etiladi.

Tahlilning uch turi mavjud:

- ❖ Eng yaxshi holat
 - Algoritm eng kam vaqt talab qiladigan kirishni belgilaydi.
 - Kirish - bu algoritmi eng tez bajariladigani.
- ❖ O'rtacha holat
 - Ishlash vaqti yoki algoritm haqida bashorat qilish imkonini beradi.
 - Kirish tasodifiy ekanligini taxmin qiladi.
- ❖ Eng yomoni
 - Algoritm uzoq vaqt talab qiladigan kirishni belgilaydi.
 - Kirish algoritmi eng sekin ishlaydi.

$$\text{lower Bound} \leq \text{Average Time} \leq \text{Upper Bound}$$

Berilgan algoritm uchun biz eng yaxshisini ifodalashimiz mumkin. Keltirilgan shaklida eng yomon va o'rtacha holatlar sifatida misol, $f(n)$ funksiya h berilgan algoritmni ifodalaydi. Bu belgi berilgan funksiyaning yuqori chegarasini beradi. $f(n) = O(g(n))$ shaklida ifodalanadi. Masalan: $f(n) = n^4 + 100n^2 + 10n + 50$. Algoritm berilgan bo'lsa, n^4 ga teng $g(n)$. Bu shuni anglatadiki, $g(n)$ kattaroq $f(n)$ uchun maksimal o'sish qoidasini beradi.



Keling, O -belgisini biroz batafsilroq ajratib ko'rsatamiz.

$O - O(g(n)) = \{f(n)$ sifatida belgilangan tushuncha: bunda c va n_0 konstantalari shunday bo'lsinki, barchasi $0 \leq f(n) \leq cg(n)$ barchasi uchun $n \geq n_0\}$ uchun asimptotik qat'iy yuqori chegara.

Bizning maqsadimiz - berilgan algoritmlarga nisbatan kattaroq yoki kichikroq bo'lgan o'sishning eng kichik yo'nalishini berishdir.

FOYDALANILGAN ADABIYOTLAR:

1. ERIC MATTHES. PYTHON CRASH COURSE PAPERBACK. ENGLAND 2015. 205 B.
2. KRISHNA RUNGTA. LEARN PYTHON IN 1 DAY: COMPLETE PYTHON GUIDE WITH EXAMPLES. INDIA 2016. -182 B.
3. Raximov S. D., Sodiqov S. S. TEXNIK SOHA MUTAXASSISLARI O 'QUV FANLARINI O 'QITISH TAYYORGARLIK JARAYONIDA C++ DASTURIDAN FOYDALANISH ZARURATI //INTERNATIONAL CONFERENCE: PROBLEMS AND SCIENTIFIC SOLUTIONS. – 2022. – T. 1. – №. 7. – С. 115-118.
4. Нам А., Саидова Ш. ПЛЮСЫ И МИНУСЫ КОМПЬЮТЕРИЗАЦИИ ПРИ ОБУЧЕНИИ ИНЖЕНЕРНОЙ ГРАФИКИ В ТЕХНИЧЕСКОМ ВУЗЕ //ОБЩЕСТВО И ИННОВАЦИИ. – 2022. – Т. 3. – №. 4/S. – С. 338-343.
5. Нам, А. Л., & Рахимов, С. (2021). НЕОБХОДИМОСТЬ ИСПОЛЬЗОВАНИЯ СИСТЕМЫ МАТЛАВ В ПРЕПОДАВАНИИ УЧЕБНЫХ ДИСЦИПЛИН ПРИ ПОДГОТОВКЕ СПЕЦИАЛИСТОВ ТЕХНИЧЕСКИХ НАПРАВЛЕНИЙ. CENTRAL ASIAN JOURNAL OF THEORETICAL & APPLIED SCIENCES, 2(5), 160-164.