

به نام خدا

# برنامه ریزی برای المپیاد کامپیوتر

(زمان بندی، پیش نیازها، منابع)

محمدجواد حاجی علیخانی

1391

## مقدمه

برای آشنایی دانش آموزان مراکز استعداد های درخشان اراک با المپیاد کامپیوتر تصمیم گرفتیم نوشته ای بنویسم که راهنمایی کوچک برای آنان باشد. معمولاً در شهرستان ها کسانی که می خواهند المپیاد بخوانند دیدروشنی نسبت به آنچه باید یاد بگیرند و چه طور باید این کار را انجام دهند ندارند، که این مشکل از نبود برنامه ی کافی و منسجم برای المپیادهای علمی است. اگر برای این کار به درستی برنامه ریزی شود، شاهد نتایج درخشانی خواهیم بود که کمترین آن ها موفقیت دانش آموزان در المپیاد است، و المپیاد بهانه ای می شود برای دنبال کردن علاقه مندی ها و استعدادها و سرآغازی بر موفقیت های دیگر.

## توضیح برنامه:

المپیاد کامپیوتر شاید یکی از مفهومی ترین المپیادها باشد. مباحث موجود در این المپیاد معمولا در مدرسه آموزش داده نمی شود. ولی برای قبولی در مرحله اول و دوم نیازی به دانش برنامه نویسی - کامپیوتر- نیست. اغلب سوال ها و مباحث مطرح شده در مرحله اول و دوم فکری و به نوعی سنجش مهارت حل مساله است. بنابراین تقویت مهارت حل مساله (که فقط با حل مساله های متعدد به دست می آید) مهم ترین بحث در المپیاد کامپیوتر است. یعنی در المپیاد کامپیوتر ما درست نگاه کردن به مساله و هجوم بردن به مساله از زوایای متعدد را یاد می گیریم.

یک برنامه ی عمومی برای این المپیاد شروع با مباحث ترکیبی است، ادامه دادن با گراف، الگوریتم و در نهایت برنامه نویسی است. ولی این برنامه با توجه به وقت و سرعت افراد می تواند فرق کند:

**توصیه های کلی:** یک موضوع بسیار مهم این است که تا جایی که میتوانید همه ی مسائل را خودتان حل کنید. حتی اگر بعد فکر کردن زیاد نتیجه ای حاصل نشد، آن سوال را علامت گذاشته و بعدا به سراغ آن بروید. یک سوال که خودتان حل کنید از ده سوال که راهش را بلد باشید با ارزش تر است.

اگر از کارکردن زیاد روز یک موضوع خسته می شوید به طور نسبتا پراکنده مساله حل کنید. مثلا دو مساله از استقرا، چند مساله از ناوردایی و... حل کنید. بازده کاری هم با این روش افزایش می یابد.

**سال اول دبیرستان:** دانش آموزان سال اول اگر به درستی برنامه ریزی کنند وقت کافی برای خواندن همه ی مباحث و حل مساله کافی خواهند داشت. برای شروع باید چیزهایی از شمارش یاد گرفت (که در مرحله اول مطرح می شود). کتاب رطخضیات انتخاب (نیمه ی اول کتاب) و الفبای المپیاد ریاضی (فصل اول) برای یادگیری شمارش مناسبند. برای تمرین بیشتر فصل های شمارشی کتاب ترکیبیات علیپور و تمرینات آن نیز مناسبند. بعد از تمام کردن شمارش - و حل مساله های کافی در این زمینه - دانش آموز باید مباحث اصل ناوردایی، لانه کبوتری و اکسترمال را فرا بگیرد. بهترین کتاب هم در این زمینه استراتژی های حل مساله است. که برای هر یک از این مباحث باید فصل مربوط به آن خوانده شود و مسائل آن هم حل شود (به جز مساله هایی که خیلی زمینه ی ریاضی دارند، مخصوصا مسائل فصل لانه کبوتری استراتژی). فصل لانه کبوتری کتاب ترکیبیات علیپور هم منبع بسیار مناسبی است.

بعد از این مباحث به یکی از مهم ترین مباحث مطرح در المپیاد کامپیوتر می رسیم، یعنی استقرا. که باید وقت کافی بر روی آن گذاشته شود. تقریبا نیمه (شاید هم بیشتر) از مساله های مرحله دو از مبحث استقراست. برای شروع فصل آخر کتاب الفبای المپیاد ریاضی یکی از کامل ترین منابع است. فصل استقرای کتاب ترکیبیات علیپور و هم چنین کتاب استراتژی های حل مساله هم مساله های خوبی دارند. انتظار می رود تا عید مباحث استقرا نیز تمام شده باشد. (شروع از ابتدای سال)

وقتی مباحث اولیه تمام شد وقت مساله حل کردن است. بیشترین وقت در این قسمت باید صرف شود. کتاب المپیادهای ریاضی شوروی یک منبع خوب مساله است. البته فقط سوالات ترکیبیاتی آن. کتاب المپیاد های ریاضی لنینگراد هم بعضا مساله های خوبی دارد. انتهای کتاب مساله های الگوریتمی هم حاوی مسائل مناسبی است که کمی سخت و پیشرفته اند. نکته مهم در این بخش حل مساله های زیاد است و نه دیدن مساله های زیاد. مساله را-هرچند سخت- باید خود فرد حل کند-همان طوری که از دیدن شنای دیگران +شنای شما قوی نمی شود.

هم زمان با حل مساله باید مقدمات نظریه گراف و سپس الگوریتم نیز فرا گرفته شود. برای نظریه گراف کتاب "نظریه گراف" نوشته داگلاس بی وست بهترین کتاب موجود است که ترجمه آن هم در بازار هست. فصل های 1-2-2 کتاب وست برای یادگیری کلی گراف کافیست. این کتاب هم چنین مساله های خوبی دارد که حتما باید حل شود. دانش آموز باید با توجه به سرعتش در حل مسایل برنامه ای بریزد که تا قبل مرحله دو حداقل 2 فصل از کتاب وست (و همچنین قضیه های مهم بخش تطابق) را بخواند.

در تابستان ورود به سال دوم هم خوب است آرام آرام الگوریتم و برنامه نویسی هم شروع شود. برای الگوریتم اولین کتاب "آشنایی با الگوریتم با رویکردی خلاقانه" معروف به creative است. الگوریتم باید به دقت خوانده شود و بهتر است کسی که قبلا این مباحث را خوانده است به دانش آموز کمک کند، گروهی هم می توان خواند. در نگاه اول بعضی الگوریتم ها سختند ولی آرام آرام در ذهن جای می گیرند.

برای شروع الگوریتم ابتدا فصل اول و دوم و پنجم کتاب creative خوانده شود. سپس باید فصل های 3-4-6-7 خوانده شود که نیاز به یک مطالعه جدی دارد. همزمان با اتمام هر فصل مسیله های آن فصل هم باید حل شود. سپس از کتاب CLRS فصل های برنامه ریزی پویا و حریصانه خوانده شود. اگر دانش آموز این مباحث را تمام کرد می تواند به سراغ الگوریتم های پیشرفته تر برود. (الگوریتم های هندسی و فصول دیگر کتاب CLRS)

**نکته:** بعضی الگوریتم ها در کلاس خیلی بهتر فهمیده می شوند. برای یادگیری الگوریتم های مهم و پایه ای دانش آموز خوب است از یک معلم یا حداقل از فرد دیگری که او هم در حال مطالعه است کمک بگیرد

برای یادگیری اصول برنامه نویسی کتاب "چگونه با ++C برنامه بنویسیم" نوشته ی دایتل و دایتل (معروف به کتاب دایتل) مناسب است. کتاب ++C جعفر نژادقمی هم کتاب روان و خوبی است (ولی ناقص است). مهم ترین چیز در برنامه نویسی تمرین زیاد است. یک معلم خوب هم می تواند خیلی کمک کند. بعد از یادگیری اصول کلی باید جدی برنامه نویسی را شروع کرد. برنامه نویسی المپیادی بیشتر حل مسئله و بیان الگوریتم به زبان برنامه نویسی است. سایت USACO سایت معروف و خوبی در این زمینه است که تعدادی سوال برای حل کردن دارد و با حل هر سوال سوال های جدیدی آورده می شود. سایت SGU هم سایت خوبی است که شامل تعداد زیادی مسئله است. سایت های timus, zoj و.. نیز مناسبند. بهتر است ابتدا 1-2 فصل از USACO حل شود سپس در سایت های دیگر شروع به برنامه نویسی کنند. سایت

madrese.inoi.ir هم سائیتی به زبان فارسی است که اکثر مسائل برنامه نویسی دوره های المپیاد کامپیوتر پیشین را هم دارد.

برنامه نویسی مبحثی است که بلید برای تسلط بر آن حوصله به خرج داد. هیچ کس یک شبه برنامه نویس نمی شود! در برنامه نویسی سه چیز مهم است: حوصله و پیگیری، تلاش و استفاده درست از منابع. یک نکته هم اینکه چون تقریباً همه ی منابع حل سؤال برای برنامه نویسی به زبان انگلیسی هستند دانش آموز باید در خواندن متون توانایی کافی داشته باشد (بعد از مدتی خواندن متن ها این توانایی به دست می آید).

در چند سال اخیر هم بعد از مرحله دو به المپیاد کامپیوتر مرحله سه اضافه شده که حضوری است و تعدادی سؤال کوتاه پاسخ برنامه نویسی دارد. برای تسلط به سؤال های کوتاه پاسخ می توان از سایت [projecteuler.net](http://projecteuler.net) کمک گرفت (حداقل 70-80 سؤال از آن حل شود).

اگر کسی این مراحل را به درستی انجام دهد، تقریباً پیش نیاز کافی برای حل همه ی مسائلی (مرحله 1، مرحله دو، دوره تابستانی) را دارد.

**نکته:** دانش آموز باید قبل از مرحله اول سوالات دوره های پیشین را از خود امتحان بگیرد. قبل مرحله دوم (حدود 1-2 هفته قبل از مرحله دوم) هم سوالات مرحله دو سال های پیش. اگر دانش آموز احساس کرد که در بحث تئوری قوی شده می تواند به سراغ سوالات فاینال تئوری دوره های قبل برود. فاینال های برنامه نویسی هم در سایت [madrese.inoi.ir](http://madrese.inoi.ir) اکثراً موجود است. البته سراغ هر سؤال باید وقتی رفت که بتوان آن را حل کرد. سوالات فاینال تئوری سال های پیش هم برای آمادگی برای مرحله دو و قوی تر شدن حل مسئله مناسب است. بعد از مرحله دو اگر دانش آموز احساس می کند می تواند قبول شود تا تابستان وقت خود را روی برنامه نویسی (و فقط برنامه نویسی) بگذارد (معمولاً کف مرحله دو پایین است چون حدود 70-80 نفر در آن قبول می شوند). سایت های [sgu](http://sgu) و [timus](http://timus) و [madrese](http://madrese) بسیار مناسبند.

**دانش آموزان سال دوم:** دانش آموزان سال دوم باید همان برنامه بالا را با فشردگی بیشتری انجام دهند. به این صورت که باید مباحث اولیه تئوری را سریع تر تمام کنند و بعد از مرحله اول برنامه نویسی را شروع کنند. (یا حداقل بعد از عید) اگر سال دومی ها بتوانند به دوره راه پیدا کنند خیلی خوب است و تقریباً بیشتر مباحث مورد نیاز را یاد می گیرند.

**سال سوم:** کسانی که از سال سوم شروع به خواندن المپیاد می کنند باید برنامه نویسی و تئوری را همراه با هم پیش ببرند. و با یک مقدار تلاش بیشتر هم مساله تئوری حل کنند و هم برنامه نویسی کنند. برنامه کلی هم همان برنامه بالاست ولی چیزهای تکمیلی مثل (ریاضیات انتخاب یا...) نیاز نیستند.

## سرفصل ها

### مباحث تئوری:

#### • ترکیبیات:

۱. شمارش: اصول جمع و ضرب، جایگشت ها، ترکیب و تبدیل
۲. اصل اکسترمال
۳. اصل ناوردابی
۴. رنگ آمیزی
۵. اصل استقرا: استقرای ضعیف، استقرای قوی، استقرای قهقراپی
۶. دو گونه شمردن (شمارش مضاعف)
۷. اصل لانه کبوتری
۸. روابط بازگشتی

#### • نظریه گراف

۱. تعاریف اولیه: راس، یال، مسیر، گشت، گذر، مولفه همبندی
۲. مسیر ها
۳. درجه رئوس، قضیه منتل، قضیه توران و دنباله های گرافیکی
۴. گراف های جهت دار و تورنمنت ها
۵. درخت و قضیه های مربوط به آن
۶. گراف های اویلری
۷. قضیه هال
۸. پوشش یالی، پوشش راسی، مجموعه های مستقل
۹. قضیه تات، قضیه کونینگ و قضیه پترسن
۱۰.  $k$  همبندی ظلی و راسی
۱۱. رنگ آمیزی یالی و قضیه ویزینگ
۱۲. رنگ آمیزی راسی و دنباله های رنگ آمیزی

۱۳. دور های همیلتونی

۱۴. برش های یالی و راسی

• الگوریتم:

۱. تحلیل الگوریتم ها:

✓ نماد های O ، امگا و تتا

✓ روش جایگذاری

✓ درخت بازگشتی

✓ فرمول اصلی

✓ تحلیل سرشکن شده

۲. آشنایی با الگوریتم

✓ مساله ستاره ی مشهور

✓ مساله نمای افقی

✓ الگوریتم هورنر

۳. ساختمان های داده ای

✓ آرایه ها

✓ لیست پیوندی

✓ بردار (آرایه ی پویا)

✓ پشته

✓ صف

✓ درخت دودویی جست و جو

✓ هیپ(هرم)

✓ Disjoint set

✓ طراحی ساختمان های داده ای

۴. مرتب سازی

✓ مرتب سازی درجی

✓ مرتب سازی هرمی

✓ مرتب سازی ادغامی

✓ مرتب سازی سریع

✓ مرتب سازی های غیر مقایسه ای

✓ مرتبه ی آماری و الگوریتم Select

✓ یافتن بیشینه و کمینه

✓ اعداد تصادفی

۵. الگوریتم های دنباله ها(غیر از مرتب سازی):

✓ جست و جوی دودویی و انواع آن

✓ تطابق رشته ای: الگوریتم های KMP و Hash

✓ کد هافمن

✓ فاصله ی ویرایشی دو دنباله

✓ یافتن اکثریت

✓ بزرگترین زیردنباله ی صعودی(LIS)

۶. الگوریتم های گراف

✓ ذخیره سازی گراف

✓ DFS

✓ BFS

✓ ساخت درخت DFS و BFS

✓ ترتیب توپولوژیک

✓ درخت پوشای کمینه

✓ الگوریتم دایسترا

✓ الگوریتم فلویید

✓ تجزیه گراف به مولفه های قویا همبند

✓ 2SAT

✓ تجزیه به مولفه های دو همبند

✓ تطابق دو بخشی

✓ LCA(اولین جد مشترک)

✓ پیدا کردن راس ها و یال های برشی

۷. برنامه ریزی پویا

✓ بزرگترین زیردنباله مشترک (LCS)

✓ ضرب زنجیر ماتریس ها

✓ عناصر روش برنامه ریزی پویا



- ✓ روش از بالا به پایین و روش پایین به بالا
- ✓ گراف زیرمسئله ها
- ✓ مسئله کوله پشتی

#### ۸. الگوریتم های حریمانه

- ✓ اثبات های حریمانه بودن
- ✓ رنگ آمیزی بازه ها
- ✓ کوله پشتی کسری
- ✓ مسئله انتخاب فعالیت

#### ۹. الگوریتم های هندسی

- ✓ ضرب خارجی و ضرب داخلی دو بردار
- ✓ محاسبه طول پاره خط
- ✓ محل برخورد دو پاره خط
- ✓ مساحت چندضلعی
- ✓ مسئله نقطه و چندضلعی
- ✓ پوش محدب
- ✓ دایره و پاره خط

#### ۱۰. NP کامل

- ✓ اثبات های NP-کامل بودن
- ✓ تحویل مساله ها به همدیگر

### برنامه نویسی:

#### • زبان C++ :

۱. برنامه نویسی چیست؟

۲. سرفایل ها

۳. متغیر ها و عملیات ریاضی

۴. دستورات ورودی/خروجی

۵. دستور های کنترلی :

✓ دستور شرطی if

✓ حلقه های for و while

✓ عملگرهای منطقی

✓ Continue,break,goto

۶. توابع

✓ توابع ریاضی cmath

✓ تعریف توابع

✓ تابع بازگشتی

✓ فراخوانی با ارجاع و مقدار

۷. آرایه ها و اشاره گرها

✓ آرایه های یک بعدی و چندبعدی

✓ رفتار آرایه ها

✓ متغیر های اشاره گر

✓ اشاره گر های رشته ای

✓ توابع پردازش رشته

۸. کلاس string و توابع مفید

۹. عملگر های بیتی، struct ها

۱۰. پیش پردازنده

۱۱. کتابخانه قالب استاندارد (STL):

✓ Vector

✓ Set

✓ Map

✓ Priority\_queue

✓ Bitset

✓ List

✓ الگوریتم های مهم STL : sort,max,min و ...

۱۲. مفهوم کلاس و استفاده از آن

• تمرین عملی

## منابع

### الف) منابع مهم

- ۱) الفبای المپیاد ریاضی، مرتضی محمدآبادی، انتشارات دانش پژوهان جوان
- ۲) استراتژی های حل مساله، انتشارات مبتکران
- ۳) المپیاد های ریاضی شوروی، مترجم پرویز شهریاری
- ۴) ترکیبیات، علیرضا علیپور، انتشارات فاطمی
- ۵) المپیاد های کامپیوتر ایران از آغاز تا کنون، مراحل اول، یاسر احمدی فولادی
- ۶) المپیاد های کامپیوتر ایران، از آغاز تا کنون: مراحل دوم، یاسر احمدی فولادی
- ۷) المپیاد های ریاضی لنینگراد
- ۸) طراحی الگوریتم با رویکردی خلاقانه، یودی منیر
- ۹) مقدمه ای بر الگوریتم ها، مترجم عین اله جعفرنژاد قمی (CLRS)
- ۱۰) Deitel & Deitel, How To Program C++
- ۱۱) آموزش برنامه نویسی با C++، دایتل و دایتل، مترجم: حسن محمدی، حسین محمدی
- ۱۲) برنامه نویسی به زبان C++، عین اله جعفرنژاد قمی
- ۱۳) آشنایی با نظریه گراف، داگلاس بی. وست، نشر گسترش علوم پایه
- ۱۴) مسئله های الگوریتمی، دکتر محمد قدسی
- ۱۵) سایت [madrese.inoi.ir](http://madrese.inoi.ir)
- ۱۶) [Acm.sgu.ru](http://Acm.sgu.ru)
- ۱۷) [Train.usaco.org](http://Train.usaco.org)

### ب) منابع تکمیلی

- ۱) جلوه هایی از ترکیبیات
- ۲) تورنمنت شهرها
- ۳) معماهای الگوریتمی، یاشار گنجعلی
- ۴) نظریه گراف و کاربرد های آن، باندی مورتی

یا حق

[m.monajem@gmail.com](mailto:m.monajem@gmail.com)