A Way to Practice Competitive Programming

- From Rating 1000 to 2400+ -

May 7th, 2019

Masataka Yoneda / E869120

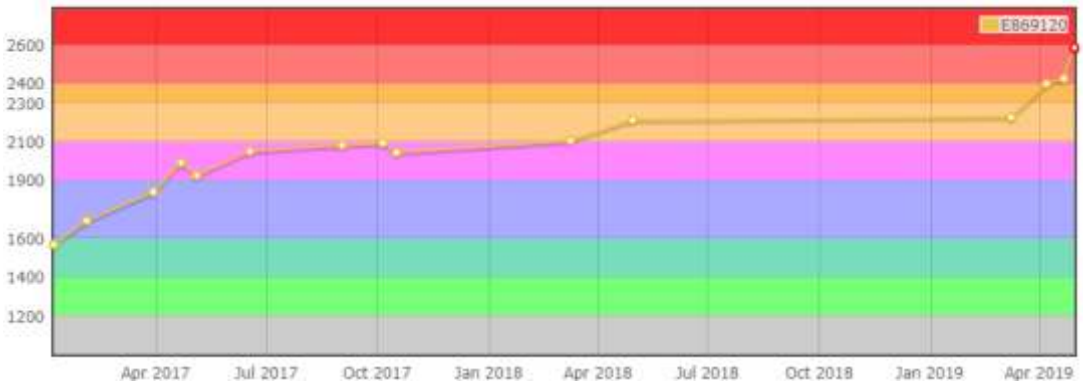
0. INDEX

- 0. Index …… p. 2
- 1. Introduction …… p. 3
- 2. A knowledge – some contest systems …… p. 5
- 3-0. Introduction to practice method …… p. 8
- 3-1. Practice Skills - From Rating 1000 to 1400 …… p. 9
- 3-2. Practice Skills - From Rating 1400 to 1900 …… p. 11
- 3-3. Practice Skills - From Rating 1900 to 2200 …… p. 13
- 3-4. Practice Skills - From Rating 2200 to 2400 …… p. 14
- 4. Practice Mental …… p. 16
- 5. A recommendation for Virtual Contests …… p. 17
- 6. Future Vision …… p. 18
- 7. Conclusion …… p. 19

1. Introduction

Dear Codeforces community.

On April 20, 2019, finally I reached a grandmaster with rating 2426, by participating 14 contests. Before last contest, I was very regretful because my rating was 2399, only 1 to become a red. So last contest (Forethought Future Cup Elimination Round) was one of my most memorable contests ever. Thank you for Codeforces to hold such contests, and thank you for kind, great and impressive Codeforces community for accepting my blog.



Today, I want to contribute to Codeforces community, because I received much benefit from Codeforces and I appreciate for them. I thought about "how to contribute to Codeforces", or the proper way to contribute to Codeforces, for long hours. I found a way: to tell and explain a way of practice from a beginner to a grandmaster.

Today, I will explain how to become a red-ranked coder.

In July 2017, I wrote a blog that explains how to become purple. (http://codeforces.com/blog/entry/53341) Though it gets 181 upvotes, there are many grammatical mistakes and some points that were insufficient for explaining. I'm sorry for this because I am not good at English and not a good writer. This time there may be some mistakes, but I made efforts to be easier to read, and to be easier to understand. Thank you for your corporation.

This time, I will explain in some steps, because there are many rating-ranges, and most

people cannot reach a grandmaster from a beginner in a single step. So do I. So, this time, I broke down into 4 parts as follows:

- Rating 1000 → 1400, from a beginner to a specialist
- Rating 1400 → 1900, from a specialist to a top 10% coder
- Rating 1900 → 2200, from a top 10% coder to a well-played Div1 player
- Rating 2200 → 2400, from a well-played Div1 player to red

In addition, before explaining about how to practice, since there are many people in Codeforces and some do not understand about other online judge systems, I'd like to explain about other contest sites. (AtCoder, TopCoder, etc.)

Again, I am very glad to be read this blog. Thank you for Codeforces.

Masataka Yoneda

2. A knowledge – Some kinds of contest sites

Recently, there are many online judges in the world. To be a grandmaster, I solved many problems from many online judges. This time, to explain the post sections, I'd like to explain about some kind of contests.

2-1. Codeforces [http://codeforces.com/]

Codeforces is one of the most famous online judges in the world. Currently, there are 5000+ problems.

One of the best ways to use Codeforces is "Problem Rating". For example, the rating of problem 1146H – Satanic Panic is 2800, which means you should solve **in the contest** if you are rating 2800 or more.

1146H	Satanic Panic	dp, geometry	2800	x103
1146G	Zoning Restrictions	dp, flows, graphs	2600	x185
1146F	Leaf Partition	dp, trees	2400	x270
1146E	Hot is Cold	bitmasks, data structures, divide and conquer, implementation	2400	x354
1146D	Frog Jumping	dfs and similar, math, number theory	2100	x595
1146C	Tree Diameter	bitmasks, graphs	1700	x1734
1146B	Hate "A"	implementation, strings	1100	x3354
1146A	Love "A"	implementation, strings	600	x4500

In addition, there are three division in Codeforces (Div1, Div2, Div3).

After next section, I will use these two representations:

- **Div1A, Div2E, etc**: Div1A means problem A of Div1 contest. Div2E means problem E of Div2 contest. Usually, the first problem is the easiest, and the last problem is the hardest.
- **Difficulty R2800 etc**: "Difficulty R2800" means that problem difficulty which tagged 2800. For example, 1146G – Zoning Restriction is difficulty R2600, 1146A – Love "A" is difficulty R600.

2-2. AtCoder [https://atcoder.jp/]

AtCoder is a contest site that **there are many good problems**. It is only "my opinion", but if you want to be better, you should solve AtCoder.

There are 3 types of contests in AtCoder:

- AtCoder Beginner Contest (ABC): There are 4 problems.
  - The **difficulty** in ABC is generally R500 – R700 – R900 – R1400 in Codeforces difficulty.

- AtCoder Regular Contest (ARC): There are 4 problems. Usually, the first two problems of ARC and the last two problems of ABC are the same.
  - The **difficulty** in ARC is generally R900 – R1400 – R2100 – R2600 in Codeforces difficulty, but it may differ by contest.

- AtCoder Grand Contest (AGC): There are 6 problems.
  - The **difficulty** in AGC is generally R1200 – R1900 – R2200 – R2500 – R3000 – R3300 in Codeforces difficulty,

There is a convenient website – AtCoder problems. (https://kenkoooo.com/atcoder#/table//)

In this website, you can know list of problems and what problem did you solve or not. Solved problems are green and attempted problems are yellow.

2-3. TopCoder

TopCoder is also a famous contest site. Single Round Match (SRM) has two divisions:

- **Division 2**: There are 3 problems and difficulties are R600 – R1300 – R2100 in Codeforces difficulty. Each problem is called Div2Easy, Div2Medium, Div2Hard.
- **Division 1**: There are 3 problems and difficulties are R1800 – R2400 – R3000 in Codeforces difficulty. Each problem is called Div1Easy, Div1Medium, Div1Hard.

I also use TopCoder for training, and to become a red-ranked coder. Though the problems are mainly math-heavy, but the problems (especially I like SRM600-699) are good.

2-4. Conclusion of this part

I mainly used 3 online judges: Codeforces, AtCoder and TopCoder. If you know 3 online judges, you are ready to read next section. Training in these contest sites is very important and effective, and I think that it is the fastest way to improve.

From part 3, I will write many essential points. But this is only **my opinion**, and it is OK to not to trust me. There are many people who followed my way and improved much, but there are also far many people who did not follow my way and improved much. Use my method to practice is good, but it is not true that my way is effective for every people.

3-0. Introduction to practice method

There may be many people who read my previous tutorial blog. But now and 2 years ago is different. There are a lot of things that I regret in 2 years. There are a lot of things that I did not do but I'm thinking that if I did this way, I could improve faster. So, there can be many **different points** between my previous blog and this blog.

In addition, my practice method will be completely different before rating 2200 and after rating 2200.

I want to say one more thing. This blog is only my opinion, and ways to practice is differ by people. So, I don't think you must do in this way. But I think that this way is one of the most effective way for some people.

If you understand this, you are ready to read. Go to the essence… 3, 2, 1, GO!

3-1. Practice Skills – From Rating 1000 to 1400

There are many gray/green coders in Codeforces. They are wanting to be a cyan/blue coder, but actually struggling for it.

Despite this fact, only three things are needed for being rating 1400.

- You can write straight-forward simulation fast. (within 5-10 minutes)
- You can write brute force fast. (within 5-10 minutes)
- You can divide the problem into some cases in your brain or in your paper. (e.g. N=2, N=3, or N>=4)

For example, in Codeforces Round #556, if you can do these three things above, surprisingly you can get even 200th place in Div2. This is very exaggerated example, but in Codeforces Round #554 (Div. 2), you can also get 3400th place, which participants whose rating less than 1250 increases rating.

On average, if you can do three things above, your rating will reach 1400.

[[How to practice]]

At first, I suggest you to solve **AtCoder Beginner Contests**.

Though there are many good problems in Codeforces, but if you want to practice to code more simply, you should solve AtCoder.

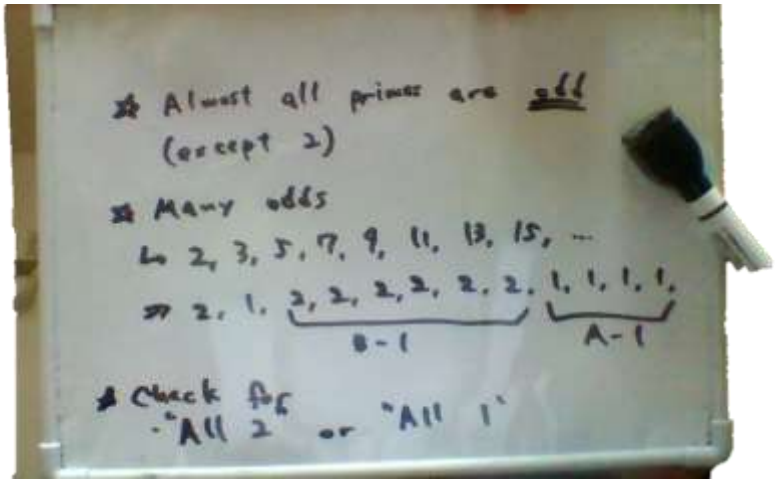
Especially, I recommend to solve problem B or C in AtCoder Beginner Contest. In problem B, you can learn how to write simulations and brute force faster. In problem C, you can learn how to consider the solution and the efficient way to use paper (or notebook) to think about solution.

From AtCoder Beginner Contest 042, English problem statement is added. Since the newest contest is AtCoder Beginner Contest 125, there are 84 Beginner Contests in AtCoder. If you solve all problems of B and C, you would learn much and you will be much stronger.

To solve problems in AtCoder, you can use AtCoder Problems, in which you can know what problem did you solve.

There are some important points when you solve problems in AtCoder.

- If you cannot reach the solution, you should read editorial after 15 minutes' thinking if problem B, and 30 minutes' thinking if problem C. Sadly, in recent ABCs, there could be no English editorial. But you can read sample solution (there are some links of sample code with high probability) to know how to solve this problem.
- Even you can solve a problem, until you get used to code faster, there may be something to learn if you look some source codes written by high-rated coders. So, I recommend to look some simple source codes.
- Especially, when you solve problem C, I recommend you to use paper to consider. For example, in Codeforces Round #556 (Div.2) in C, you can write memo like this:



About this is not paper but whiteboard. Using whiteboard instead of paper is also good.

Since I am a leader of computer club in a high-school, so I have many clubmates, and there are a lot of clubmates which uses this method and improved.

3-2. Practice Skills – From Rating 1400 to 1900

One of the most major rating range in Codeforces is [1400, 1500]. They are wanting to improve rating, but from rating 1500 is difficult and many people gave up here. But there are also many people who practiced without giving up and gained rating.

To be rating 1900, skills as follows are needed:

- You know and can use major algorithms like these:

Brute force	DP	DFS	BFS	Dijkstra
Binary Indexed Tree	nCr, nPr	Mod inverse	Bitmasks	Binary Search

Note: I think that segment tree is not needed to be rating 1800. I knew segment tree when I was a purple-rated coder.

- You can code faster (For example, 5 minutes for R1100 problems, 10 minutes for R1400 problems) Fast coding is very important for Codeforces because generally, if there is a wide gap of difficulty in a contest, fast coding affects rating very much.

[[How to Practice]]

If you are not good at fast-coding and fast-debugging, you should solve AtCoder problems. Actually, and statistically, many Japanese are good at fast-coding relatively while not so good at solving difficult problems. I think that's because of AtCoder.

I recommend to solve problem C and D in AtCoder Beginner Contest. **On average**, if you can solve problem C of AtCoder Beginner Contest within 10 minutes and problem D within 20 minutes, you are Div1 in FastCodingForces :)

If you are not good at solving more than R1400 in Codeforces, you should learn some algorithms above and solve typical problems in Codeforces. For example, if you felt that you are not good at DP, solve DP-tagged problem in Codeforces which is R1200-R1400. Surprisingly, there are only ~50 problems which tagged DP and less than or equal to R1400.

Interestingly, typical problems are concentrated in Div2-only round problems. If you are not good at Div2-only round, it is likely that you are not good at using typical algorithms, especially 10 algorithms that are written above.

If you can use some typical problem but not good at solving more than R1500 in Codeforces, you should begin TopCoder. This type of practice is effective for people who are good at Div.2 only round but not good at Div.1+Div.2 combined or Div.1+Div.2 separated round.

Sometimes, especially in Div1+Div2 round, some problems need mathematical concepts or thinking. Since there are a lot of problems which uses them (and also light-implementation!) in TopCoder, you should solve TopCoder problems.

I recommend to solve Div1Easy of recent 100 SRMs. But some problems are really difficult, (e.g. even red-ranked coder could not solve) so before you solve, you should check how many percent of people did solve this problem. You can use https://competitiveprogramming.info/ to know some informations.

Unfortunately, I don't know the good website which you can know what problem did you solve in TopCoder SRM, like AtCoder Problems. So, if you want to record what problem did you solve, you should make spreadsheets or tables. This is my brother's spreadsheets which he used two years ago. (Sorry for Japanese, but there are no Japanese in Div1Medium page, so you can refer to it)

https://drive.google.com/file/d/1mSy9PM4Km8EVv8Lp4nhitorOe2HbAS1e/view?usp=sharing

When I was a blue, I was also very bad at mathematical thinking. When I solve 50 Div1Easys, I became yellow in TopCoder and purple in Codeforces.

If you are good at solving problems but did not perform well in real contests, you should do virtual contests more. Do you know Virtual Contest system in Codeforces? You can do virtual participation!

3-3. Practice Skills – From Rating 1900 to 2200

If you want to be rating 2200, at first, you should be Div1 and compete in Div1 contests. It means you should solve a lot of difficult problems. (e.g. R1900 or more in Codeforces)

Even if you are good at fast-solving or solving typical problems very much, competing in Div1 is very different. Sadly, there was a lot of coders which travelling between blue and purple.

To be rating 2200, skills as follows are needed:

- You know and can use 10 algorithms which I stated in pp.11 and segment trees (including lazy propagations)
- You can solve problems very fast: For example, 5 mins for R1100, 10 mins for R1500, 15 mins for R1800, 40 mins for R2000.
- You have decent skills for mathematical-thinking or considering problems
- Strong mental which can think about the solution more than 1 hours, and don't give up even if you are below average in Div1 in the middle of the contest

[[How to Practice]]

This is only my way to practice, but I did many virtual contests when I was rating 2000. In this page, virtual contest does not mean "Virtual Participation" in Codeforces. It means choosing 4 or 5 problems which the difficulty is near your rating (For example, if you are rating 2000, choose R2000 problems in Codeforces) and solve them within 2 hours.

You can use https://vjudge.net/. In this website, you can make virtual contests from problems on many online judges. (e.g. AtCoder, Codeforces, Hackerrank, Codechef, POJ, …)

If you cannot solve problem within the virtual contests and could not be able to find the solution during the contest, you should read editorial. Google it. (e.g. If you want to know editorial of Codeforces Round #556 (Div. 1), search "Codeforces Round #556 editorial" in google)

There is one more important thing to gain rating in Codeforces. To solve problem fast, you should equip some coding library (or template code). For example, I think that equipping segment tree libraries, lazy segment tree libraries, modint library, FFT library, geometry library, etc. is very effective.

3-4. Practice Skills – From Rating 2200 to 2400

This is the last part of practice skills in this blog. Actually, I had been orange for long time, and also in virtual contests my average performance had been orange for a long time. That's because my practice method that I did before got stuck when I reached orange.

Rating 2200 and 2400 is actually very different – if you have average performance of rating 2200, if you participate contest more, touch at red-rating (it means reaching max rating 2400) seems like not hard. But getting average performance of rating 2400 is much more difficult than you think. If your rating is exactly 2400, in Div.1 contest, generally you should get around 80-percentile rank (e.g. If 525 people participated, you should get 105th) to gain 1 rating or more.

To be rating 2400, skills as follows are needed:

- You should have skills that stated in previous section (rating 2200)
- You should solve **difficult problems** which are only solved by less than 100 people in Div1 contests

If you want to solve difficult problems, ad-hoc problems. According to TozanSoutherPacks' comment from my previous tutorial blog (http://codeforces.com/blog/entry/53341?#comment-373965), "To reach more than 2600, you should solve boss problems and all these are ad-hoc problems or many-steps consideration problems." I think that it's true, but for me I felt like even if you want to reach rating 2400, solving **some part of** many-steps ad-hoc problems are needed.

[[How to Practice]]

The most secure way to reach 2400 is "Solve 4000 problems". I solved more than 4000 problems including TopCoder, AtCoder, Codeforces, and problems from some Japanese online judges.

Actually, there is a legend (or actually truth) that one of the greatest coder tourist solved 10000+ problems in his life.

But there may be a lot of people who think they have no time. So I will give you some effective ways.

At first, there are a lot of educational problems in AtCoder. I recommend you should solve problem E and F (especially 700-900 points problem in AtCoder) of **AtCoder Regular Contest, especially ARC058-ARC090**. Though old AtCoder Regular Contests are balanced for "considering" and "typical", but sadly, AtCoder Grand Contest and recent AtCoder Regular Contest problems are actually too biased for considering I think, so I don't recommend if your goal is gain rating in Codeforces. (Though if you want to gain rating more than 2600, you should solve problems from AtCoder Grand Contest)

For me, actually, after solving AtCoder Regular Contests, my average performance in CF virtual contest increased from 2100 to 2300 (I could not reach 2400 because start was early)

If you cannot solve problems, I recommend to give up and read editorial as follows:

Point value	600	700	800	900	1000-
CF rating	R2000	R2200	R2400	R2600	R2800
Time to editorial	40 min	50 min	60 min	70 min	80 min

If you solve AtCoder educational problems, your skills of competitive programming will be increased. But there is one more problem. Without practical skills, you rating won't increase.

So, you should do **50+ virtual participations (especially Div.1)** in Codeforces. In virtual participation, you can learn how to compete as a purple/orange-ranked coder (e.g. strategy) and how to use skills in Codeforces contests that you learned in AtCoder. **I strongly recommend to read editorial of all problems except too difficult one** (e.g. Less than 30 people solved in contest) after the virtual contest. I also recommend to write reflections about strategy, learns and improvements after reading editorial on notebooks after the contests/virtual.

In addition, about once a week, I recommend you to make time to think about much difficult problem (e.g. R2800 in Codeforces) for couple of hours. If you could not reach the solution after thinking couple of hours, I recommend you to read editorial because you can learn a lot. Solving high-level problems may give you chance to gain over 100 rating in a single contest, but also can give you chance to solve easier problems faster.

Lastly, I guess that this method works over 30% of people because I reached Codeforces red by using this method, and my twin brother square1001 reached TopCoder red by using the method extended for TopCoder. Hope new red-ranked coders will be born by using my method :)

3. Practice Mental

Problems about mental are one of the most common problem in competitors. Even for me, until just 2 months ago, because of the fear of dropping 100+ rating in a single contest, I could not participate in any rated CF contest for more than 9 months. Also, there were several contests that I could not even read latter problems because I could not solve easier problem and lost my cool.

I found a way to practice mental recently.

- Make a routine that you will do just before the contest. It directly leads to the concentration during the contest.
  - Actually, for me, my routine is to watch the countdown from very near distance to PC before the contests starts. Legendary grandmaster yutaka1999 actually sit in Zen meditation (religious meditation) before IOI competition starts.
- If you are not in good condition in contests (e.g. Cannot solve problem B within 20 minutes) not looking current standings is also a good idea.
- Another important thing is "don't care about rating during contest".
  - I think that even if my rating dropped, the probability that I gain rating next time increases. That's why I don't care about rating **during contest**.
- Don't care about mistakes. Mistakes are also a good chance to learn, and there is no person who don't make any mistake. Even tourist made a mistake in AtCoder World Tour Finals.
  - I think that the there is a correlation between "rating" and "how many mistakes did you made **and reviewed**".

These method above works not only in contests, but also in virtual participations. Be careful: do not have too much pressure in contests and have fun!

4. Recommendation of virtual contests

Before writing about recommendation of virtual contests, I think that there are two types of virtual contests as follows:

- Virtual Participation: Choose the contests and solve problems as if it is a real contest. As same as Codeforces Virtual Participation system.
- Virtual Contest: Use https://vjudge.net/ or some other judges. Choose some problems which fit your rating from Codeforces, and upsolve within 2-3 hours.

At first, I will recommend both types of virtual contests.

For virtual participation, I think that you can learn about strategy, mental, and some other kinds of contest skills. For virtual contest, I think that you can concentrate more than solving single problem, so the efficiency of solving/reviewing will increase.

That's why when I upsolve some difficult problems in Codeforces, I always use https://vjudge.net/ and make virtual contests.

If you want to do virtual contests in AtCoder, use https://not-522.appspot.com/ (sadly it is Japanese, use google translate if you are not Japanese sorry…)

5. Future Vision

Luckily, at April 29th, 2019, I reached rating 2585 in Codeforces and it was very close to being International Grandmaster. I'm very glad about it and I really appreciate for Codeforces admins and Codeforces community.

Contests

#	Contest	Rank	Solved	Rating change	New rating	
15	Codeforces Round #556 (Div. 1)	15	3	+159	2585	
14	Forethought Future Cup - Elimination Round	89	5	+27	2426	Became Grandmaster E869120 → E869120
13	Codeforces Global Round 2	30	5	+173	2399	Became International master
12	Codeforces Round #545 (Div. 1)	154	2	+14	2226	
11	Codeforces Round #477 (rated, Div. 1, based on VK Cup 2018 Round 3)	63	3	+107	2212	E869120 → E869120
10	Codeforces Round #469 (Div. 1)	130	3	+61	2105	Became Candidate Master
9	Codeforces Round #441 (Div. 1, by Moscow Team Olympiad)	295	2	-49	2044	

I actually determined my future goal! : Be a legendary grandmaster.

Actually, recently I am doing many contests/virtual participations in Codeforces. My recent 10 performances on CF contests including virtual are: 2865, 2661, 3029, 2890, 2858, 2875, 2318, 2717, 2506, 2891. (ave = 2761.0)

So, I should gain skills of ~250 rating to be a legendary grandmaster. What I am doing to achieve my dream is as follows:

[[How to Practice]]

- Almost every day, do Codeforces virtual participation for 2-3 hours. After the contest, read the editorial and solution of legendary grandmasters, and write reviews for 1 hours.
- Sometimes (around once a week), choose ~5 problem from Codeforces (Difficulties are R2200-R2800) and solve within 2-4 hours.
- Although I experienced competitive programming for 3 years, I think that I don't know many techniques (not actually as same as typical) so I always read editorial if I could not solve problems which had solved by >20 people during actual contest.
  - ➢ I think that if I could solve almost all problems which had solved by >20 people during the contest, I would be LGM.

6. Conclusion

That's all that I can tell about the practice method to the Codeforces community. Really thank you for reading my blog.

I'm using and participating Codeforces for a long year and grew up in this community. So, I think that I should contribute to Codeforces community for something. There are many ways to contribute, but I decided to tell and report how to practice many skills that are usable for gaining rating in Codeforces. I think that this is one of the best ways to contribute with real-meaning in Codeforces.

Finally, I suggest that steps and practice methods to improve your competitive programming skills. But ways to practice differs by person. So, I don't think that my way is always the most effective way, and also, I don't think you must do this way.

But I think that this is one of the most effective way for some people. Hope this blog is useful even a little.

Since I'm only a Japanese 11th grader, there may be many grammar mistakes and my English is poor. I have to apologize about that but conversely, I have to appreciate to the readers. Thank you for reading.

Finally, I hope good luck and high rating in your Codeforces life, and having fun in competitive programming. This is the conclusion of my blog. Thank you.

Masataka Yoneda